



Universidad de las Ciencias  
Informáticas

**Facultad de Ciencias y Tecnologías Computacionales**

**PORTAL PARA EL GRUPO  
DE USUARIOS POSTGRESQL DE CUBA**

Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas

**Autor:** Yandy Ernesto Cantillo Remis

**Tutora:** MsC. Yudisney Vazquez Ortíz

**La Habana, 22 de junio de 2017**

**Año 59 de la Revolución**

## **DECLARACIÓN DE AUTORÍA**

---

Declaro ser el único autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

---

Yandy Ernesto Cantillo Remis

Autor

---

MSc. Yudisney Vazquez Ortíz

Tutora

## DEDICATORIA

---

*Dedico esta tesis a mis padres y en especial a mi hermana. Ellos me han apoyado en toda mi trayectoria para hacerme ingeniero, así como en las decisiones que he tomado en todo este tiempo. Además, se la dedico a mi novia quien fuera mi compañera de tesis y por razones de la vida no pudo ser.*

## AGRADECIMIENTOS

---

Primero que todo quiero dar gracias a Dios por permitir que este momento llegara, a Él le debo todos mis logros. A mis padres por haberme apoyado en todo momento y decirme que sí se puede cuando menos lo creía, por guiar cada paso de mi vida como nadie sería capaz de hacerlo, gracias a ustedes hoy soy ingeniero. A mi hermana por haber sido mi guía en todo momento, por ser un ejemplo a seguir y por exigirme tanto, gracias por tu cariño y apoyarme tanto. A mi novia Patricia que deseo con todo mi corazón que seas mi futura esposa, gracias por apoyarme en todo momento y aunque estés a kilómetros de distancia no me dejaste caer en la tristeza, espero que hoy después de llorar estemos contentos, porque este logro es de los dos. A mis suegros que me permitieron de una forma u otra cubrir el vacío que dejó por unos meses su niña, por quererme como a un hijo y apoyarme en todo lo que necesité, gracias a Mary, al Muerto y a toda la familia. A mi cuñado, que desde que formamos una familia me has enseñado que no importa cuánto esté invertida la pirámide, algún día tomará su lugar y los ingenieros tendremos el lugar que merecemos. A mi tutora por apoyarme en este proceso tan importante, siempre estuviste ahí cuando lo necesité, sin hablar de la fuerza que hiciste cada vez que quería rendirme. A Yanet y a Rolando por cogerme tantos errores en el documento, al igual que Dayana por ser tan tan quisquillosa, a Grina por darme fuerzas desde primer año para engordar un poco, aunque no lo haya logrado, en fin a todos esos profesores que me ayudaron a formarme como ingeniero. A Ismary, Yazmin y al Padry, por brindarme su amistad incondicional y estar ahí cuando más los necesité. A mi profesora Star, que me enseñó que si queremos algo tenemos que luchar por ello. A mis hermanitas de nacimiento, Rosmery y Janet, gracias por apoyarme en todo este tiempo. A mis segundos padres Rosa, Tía Luisa, Rey y Tony. A mis amigos, a los que están presentes y a los que no (Migue, Damián, Dariel, Marcel, Jorgito, Denis, Chinos, a todos). A mi familia de la UCS, los jugadores de Call y Battle Morato, Alucard y Destroyer, gracias por apoyarme y ayudarme, Yuli tú no te escapas, haz sido mi hermana, mi amiga, mi pañuelo de lágrimas y eso te lo agradezco mucho, Gretel llegaste última, pero te colaste, Enier yo sé que hubieses querido ser más que

un amigo, pero esa plaza está llena mi hermano, gracias a todos por su apoyo. A todas las personas que me apoyaron de una forma u otra en la tesis y por falta de páginas no los pude mencionar, GRACIAS!!!

## **RESUMEN**

---

El desarrollo acelerado de las Tecnologías de la Información y las Comunicaciones y su incorporación definitiva a todas las esferas de la vida ha incrementado exponencialmente el volumen de datos a archivar, requiriéndose mecanismos más eficientes para almacenar y manipular estos datos; siendo las bases de datos una de las soluciones de mayor impacto. En Cuba, se apuesta por el empleo de PostgreSQL como gestor de bases de datos relacional, existiendo un grupo de usuarios que promueve su uso mediante actividades de capacitación y eventos, difundidos a través de un portal. El Grupo de Usuarios PostgreSQL de Cuba requiere, además de publicar noticias y documentación, la gestión de sus miembros, eventos y actividades de capacitación, necesidades que no son soportadas actualmente por el portal en línea. Para solventar las necesidades del Grupo se realizó la presente investigación en la que, utilizando Drupal, PHP, HTML, CSS y PostgreSQL, se obtuvo como resultado una solución que gestiona todos los procesos del Grupo desde una única localización. Dicha aplicación fue validada mediante pruebas funcionales, de rendimiento y de fiabilidad las cuales arrojaron como resultado una aplicación que cumple con las necesidades planteadas por el cliente.

Palabras clave: capacitación, comunidad, eventos, miembros, PostgreSQL.

## **ABSTRACT**

---

The accelerated development of Information and Communication Technologies and its definitive incorporation into all spheres of life has exponentially increased the volume of data to be archived, requiring more efficient mechanisms to store and manipulate this data; being the databases one of the solutions of greater impact. In Cuba, is committed PostgreSQL as relational database management system, with a User Group that promotes its use through training activities and events, spread through a portal. The PostgreSQL Users Group of Cuba requires, in addition to publishing news and documentation, the management of its members, events and training activities, that don't be supported by the actual portal. To solve this Group's requirements was realized this research where was used Drupal, PHP, HTML, CSS and PostgreSQL, obtaining as result a solution that management all Group processes from a single location.

Keywords: community, event, member, PostgreSQL, training.

# ÍNDICE DE CONTENIDOS

---

INTRODUCCIÓN .....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA .....	5
1.1. Gestión de comunidades.....	5
1.2. Portales web de las comunidades de PostgreSQL de Hispanoamérica .....	6
1.2.1. Portal web del Grupo de Usuarios PostgreSQL de Cuba.....	6
1.2.2. Portales web del Grupo de Usuarios PostgreSQL de Argentina y de la Comunidad Brasileña de PostgreSQL .....	7
1.2.3. Portal web en español sobre PostgreSQL.....	7
1.3. Metodología, tecnologías y herramientas para el desarrollo de la solución.....	8
1.3.1. Metodología de desarrollo de software.....	8
1.3.2. Sistemas de Gestión de Contenidos .....	9
1.3.3. Herramienta y lenguaje para el modelado.....	12
1.3.4. Lenguajes de programación y de diseño.....	13
1.3.5. Sistema de gestión de bases de datos.....	15
1.3.6. Servidor web.....	16
1.3.7. IDE de desarrollo NetBeans .....	18
Conclusiones del capítulo .....	18
CAPÍTULO 2: ANÁLISIS Y DISEÑO DEL PORTAL PARA EL GRUPO DE USUARIOS POSTGRESQL DE CUBA .....	19
2.1. Modelo de dominio .....	19
2.2. Requisitos funcionales .....	20
2.3. Requisitos no funcionales .....	22
2.4. Análisis del sistema.....	23
2.4.1. Actores del sistema .....	23
2.4.2. Casos de uso del sistema.....	23
2.4.3. Diagrama de casos de uso del sistema.....	24
2.5. Diseño del portal para el Grupo de Usuarios PostgreSQL de Cuba .....	34
2.5.1. Arquitectura de software.....	35
2.6. Patrones .....	37
2.6.1. Patrones de diseño.....	37
Conclusiones del capítulo .....	38
CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA .....	39

3.1. Diagrama de despliegue .....	39
3.2. Vista de implementación .....	39
3.3. Diseño de la base de datos .....	43
3.4. Pantalla principal del sistema .....	44
3.5. Pruebas realizadas al sistema .....	45
3.5.1. Métodos de prueba .....	46
3.5.2. Diseño de los casos de prueba .....	47
3.5.3. Resultados de las pruebas .....	52
Conclusiones del capítulo .....	53
CONCLUSIONES GENERALES .....	54
RECOMENDACIONES .....	55
BIBLIOGRAFÍA .....	56
ANEXOS .....	62



## INTRODUCCIÓN

---

Desde sus inicios el hombre ha tenido la necesidad de guardar información, apoyándose en medios que han evolucionado desde piedras, pieles tratadas, papiros, pergaminos, hojas, cintas magnéticas, tarjetas perforadas, disquetes, CD-DVD y hasta discos duros y memorias flash hoy día.

Con el desarrollo acelerado de las Tecnologías de la Información y las Comunicaciones y su incorporación definitiva al trabajo de empresas, organismos, instituciones e incluso, a la vida personal, se ha incrementado exponencialmente el volumen de datos a archivar, requiriéndose mecanismos más eficientes para almacenar y manipular dicha información.

Una de las soluciones de mayor impacto ha sido el empleo de bases de datos, término estandarizado en la década del 60 del siglo pasado, por Date, como *“el conjunto de datos persistentes que es utilizado por los sistemas de aplicación”* (Date, 1999).

Aunque actualmente las bases de datos se basan en varios modelos; siendo aún las relacionales ampliamente utilizadas como soporte de las aplicaciones informáticas. Estas son manipuladas por un sistema de gestión de bases de datos, software de propósito general que facilita los procesos de definición, construcción y manipulación de bases de datos para diversas aplicaciones; existiendo varios sistemas comerciales que pudieran emplearse para la gestión de los datos, dependiendo su elección de las necesidades y condiciones del cliente.

En Cuba, antes del 2009, existía una amplia diversidad en el empleo de gestores, derivando en inconvenientes entre los que destacaban que los gestores propietarios, como Oracle y SQL Server, eran utilizados de manera ilegal amparados en el bloqueo económico y comercial impuesto por Estados Unidos que, de pagarlos, los montos ascenderían a miles de dólares. Independientemente del dinero a pagar para utilizarlos, existía la alta posibilidad de que en estas soluciones existieran puertas traseras. Las empresas de desarrollo de software, como la UCI, estaban imposibilitadas de comercializar productos que tuvieran como base gestores propietarios o solo podían emplearlos a través de terceros, lo cual encarecía la solución y; los gestores libres tenían el riesgo latente de dejar de serlo, como pasó con FoxPro y MySQL (Vázquez Ortiz, 2013).

Estos elementos contribuyeron a que un grupo de especialistas comenzaran a promover la utilización de PostgreSQL; en consonancia con el proceso de migración a software de código abierto del país. El Grupo de Usuarios PostgreSQL de Cuba (GUP.cu) fue creado, oficialmente, en febrero del 2009 con los propósitos de contribuir al fortalecimiento de la soberanía tecnológica cubana desde el enfoque del desarrollo de

tecnologías de bases de datos tomando como base a PostgreSQL; proveer soluciones integrales y consultorías relacionadas con la migración y la explotación de PostgreSQL y contribuir a la formación de especialistas de alto nivel en tecnologías de bases de datos (Vázquez, y otros, 2011).

El Centro de Tecnologías de Gestión de Datos (DATEC) de la Facultad de Ciencias y Tecnologías Computacionales en la Universidad de las Ciencias Informáticas (UCI), tiene un núcleo importante de dicho grupo de usuarios, y donde, para mantener comunicación con todos los miembros del GUP.cu; informar a los que no tienen Internet sobre las novedades del gestor; el desarrollo de eventos y la impartición de cursos de capacitación, se creó el portal *PostgreSQLenCuba*; pero esta solución no es factible actualmente para la gestión de todos los procesos que se llevan a cabo por el GUP.cu ya que:

- No permite la gestión de los miembros del GUP.cu, impidiendo el control de cuántas personas pudieran colaborar con determinada actividad.
- No permite la gestión de los eventos, exigiendo que la comunicación para la inscripción deba realizarse mediante correo electrónico u otra vía y llevando el control de los participantes en Excel, haciendo engorroso el proceso y pudiendo derivar en errores que afecten el desarrollo exitoso de los eventos.
- No permite la gestión de actividades de capacitación, exigiendo que las solicitudes de matrícula deban realizarse mediante correo electrónico, haciendo lento y engorroso el proceso.

A raíz de la situación existente surge como **problema de investigación**: las escasas opciones que brinda el portal *PostgreSQLenCuba*, afectan la gestión y control de las actividades llevadas a cabo en el Grupo de Usuarios PostgreSQL de Cuba.

Para darle solución se define como **objeto de estudio**: el proceso de desarrollo de aplicaciones web para comunidades virtuales de PostgreSQL; se trazó como **objetivo general**: desarrollar un portal para el Grupo de Usuarios PostgreSQL de Cuba, que facilite la realización de los procesos llevados a cabo en él; desglosándose en los siguientes **objetivos específicos**:

- Elaborar el marco teórico sobre el desarrollo de los portales de las comunidades de PostgreSQL.
- Identificar las funcionalidades del portal para que el Grupo de Usuarios PostgreSQL de Cuba cuente con una solución para la gestión de sus procesos.
- Diseñar el portal para el Grupo de Usuarios PostgreSQL de Cuba.
- Implementar el portal para el Grupo de Usuarios PostgreSQL de Cuba.
- Verificar el correcto funcionamiento del portal para el Grupo de Usuarios PostgreSQL de Cuba.

En función de alcanzar los objetivos trazados se enmarcó la investigación en el **campo de acción**: el proceso de desarrollo de aplicaciones web para comunidades virtuales de PostgreSQL en Cuba.

Para evaluar el cumplimiento de los objetivos propuestos se plantean las siguientes **preguntas científicas**:

- ¿Qué funcionalidades brindan los portales de las comunidades de Hispanoamérica?
- ¿Qué metodología, tecnologías y herramientas utilizar para el desarrollo del portal para el Grupo de Usuarios PostgreSQL de Cuba?
- ¿Qué características se deben tener en cuenta para el funcionamiento del portal para el Grupo de Usuarios PostgreSQL de Cuba?
- ¿Cómo estructurar el proceso de diseño del portal para la gestión del GUP.cu?
- ¿Cómo estructurar el proceso de implementación del portal para la gestión del GUP.cu?
- ¿Cómo validar el correcto funcionamiento del portal para la gestión del GUP.cu?

Para darle cumplimiento a los objetivos se plantean las siguientes **tareas de la investigación**:

- Análisis de soluciones de gestión de comunidades de PostgreSQL existentes para definir las principales características que debe poseer el Grupo de Usuarios PostgreSQL de Cuba.
- Definición de las herramientas, tecnologías y metodología a utilizar para el desarrollo del sistema para el Grupo de Usuarios PostgreSQL de Cuba.
- Definición de los requisitos funcionales y no funcionales para la implementación del portal para el Grupo de Usuarios PostgreSQL de Cuba.
- Realización del análisis y diseño del portal para el Grupo de Usuarios PostgreSQL de Cuba.
- Desarrollo del portal para el Grupo de Usuarios PostgreSQL de Cuba.
- Definición de los tipos de pruebas para la validación del portal para el Grupo de Usuarios PostgreSQL de Cuba.
- Aplicación de las pruebas para la verificación del portal para el Grupo de Usuarios PostgreSQL de Cuba.

Para llevar a cabo la investigación se utilizan los **métodos teóricos y empíricos**:

- Método teórico: permiten revelar las relaciones esenciales del objeto de investigación, no observables directamente, participan en la etapa de asimilación de hechos, fenómenos y procesos y en la construcción del modelo (Alvarez de Zayas, 1995). Como método teórico se determinó el empleo del Analítico-Sintético, que permite analizar los elementos comunes de los portales de las comunidades de PostgreSQL en Hispanoamérica, así como las características de las herramientas y tecnologías a emplear para elegir las más adecuadas.
- Método empírico: revela y explica las características fenomenológicas del objeto, se emplea fundamentalmente en la primera etapa de acumulación de información (Alvarez de Zayas, 1995). Se

determinó el empleo de la observación para obtener información sobre el funcionamiento de portales similares al que se va a desarrollar.

Los resultados de la investigación están plasmados en el presente informe, estructurado de la siguiente manera:

- Capítulo 1 Fundamentación teórica: en este capítulo se hace un estudio de los elementos teóricos referentes al portal para el Grupo de Usuarios PostgreSQL de Cuba, se plantean además los conceptos asociados al dominio del problema, se aborda con profundidad el objeto de estudio, la situación problemática y se analizan otras soluciones existentes; además se caracterizan las herramientas y tecnologías que se utilizarán para la construcción de la solución.
- Capítulo 2 Análisis y diseño: en este capítulo se realiza el análisis y diseño de la solución propuesta, se describe el modelo de dominio, se enumeran y detallan los requisitos funcionales y no funcionales que debe tener el portal partiendo de conocer las condiciones y cualidades deseadas, se realiza el diagrama de casos de uso del sistema y su descripción textual.
- Capítulo 3 Implementación y prueba: en este capítulo se realiza el flujo de trabajo de implementación, donde se realizan los principales artefactos como el modelo de implementación y el modelo de despliegue; además, se realizan pruebas a nivel de sistema como son las de funcionalidad, fiabilidad y rendimiento y un análisis de los resultados de las mismas, que permiten la validación de la solución propuesta.
- Conclusiones generales: donde se realiza una evaluación del cumplimiento de los objetivos de la investigación.
- Recomendaciones: donde se listan las sugerencias para el perfeccionamiento y ampliación del alcance de la investigación.
- Bibliografía: donde se reflejan todas las fuentes bibliográficas referenciadas y consultadas para realizar el informe de la investigación.
- Anexos: donde se muestra el acta de aceptación.

# CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

---

En el capítulo se analizan los portales de las comunidades de PostgreSQL en Hispanoamérica para determinar cómo dan respuesta a las necesidades del Grupo de Usuarios PostgreSQL de Cuba. Además, se realiza un estudio de las tecnologías necesarias para la implementación de la solución para el GUP.cu.

## 1.1. Gestión de comunidades

Una comunidad es un conjunto de individuos que tienen en común diversos elementos, como puede ser el idioma, el territorio que habitan, las tareas, los valores, los roles, el idioma o la religión (Concepto, 2015).

Las comunidades se establecen por un grupo de personas con objetivos comunes que utilizan una misma forma de comunicación para interactuar entre ellos y que tienen un sentimiento de pertenencia al grupo. Aquellas virtuales tendrían las características mencionadas, con la particularidad de que se desarrollaría en un entorno en línea (Turner, 2013).

Para la presente investigación se asume el concepto de Turner por centrarse en el estudio de una comunidad técnica, teniendo en cuenta que las comunidades virtuales permiten la comunicación entre sus miembros, aunque estén en localizaciones físicas distantes, siendo este concepto base para el desarrollo de la investigación.

En Cuba, y fundamentalmente en el área de la informática, han surgido varias comunidades con el propósito de fomentar la comunicación entre sus miembros, independientemente de la distancia entre ellos, y haciendo uso de las Tecnologías de la Información y las Comunicaciones.

Una de las comunidades existentes es el Grupo de Usuarios PostgreSQL de Cuba, creado en febrero del 2009 en la Universidad de las Ciencias Informáticas con especialistas de la propia universidad, ETECSA y ECASA, esencialmente. Dicho grupo fue creado en respuesta a los problemas derivados del empleo de sistemas de gestión de bases de datos propietarios como Oracle y Microsoft SQL Server, para contribuir de manera activa y decisiva con la soberanía tecnológica potenciando el uso y explotación del gestor de código abierto (Mesa, y otros, 2012).

Para cumplir con su objetivo el GUP.cu se trazó como propósitos el de contribuir al fortalecimiento de la soberanía tecnológica cubana desde el enfoque del desarrollo de tecnologías de bases de datos tomando como base a PostgreSQL, proveer soluciones integrales y consultorías relacionadas con la migración y la explotación de PostgreSQL y contribuir a la formación de especialistas de alto nivel en tecnologías de bases de datos (Vázquez, y otros, 2011).

Para ello, y para fomentar la comunicación entre sus miembros, se creó en la Universidad de las Ciencias Informáticas un portal que permite la gestión de noticias y documentación, pero que no contempla otros procesos que se llevan a cabo desde el GUP.cu y que, al no ser gestionados desde un único espacio, afectan su funcionamiento, siendo necesario una solución centralizada que permita su gestión.

## **1.2. Portales web de las comunidades de PostgreSQL de Hispanoamérica**

Hispanoamérica es una de las zonas geográficas que, con más fuerza, promociona el empleo y adopción de PostgreSQL como sistema de gestión de bases de datos. En esta área han surgido varias comunidades con este propósito, respaldadas por sitios web como vía para compartir noticias, proyectos, experiencias y temas relacionados con el gestor en sus respectivos países.

### **1.2.1. Portal web del Grupo de Usuarios PostgreSQL de Cuba**

El portal web desarrollado para el trabajo del Grupo de Usuarios PostgreSQL de Cuba, disponible en <https://postgresql.uci.cu>, fue creado con el objetivo de lograr la visibilidad del GUP.cu desde un único espacio, promover el empleo del gestor y brindar servicios de capacitación a los usuarios en el país; permite, sin necesidad de tener Internet:

- Mostrar información relacionada con la actualidad del gestor y la realización de acciones en el país asociadas a él, acciones que pueden ser desarrollo, realización de eventos, realización de acciones de capacitación.
- La descarga de manuales, guías y documentación para el trabajo con el gestor.
- La descarga de código fuente, instaladores del gestor e instaladores de herramientas para el trabajo con él.

Sin embargo, no permite:

- La gestión de los miembros del GUP.cu, no pudiendo contabilizarse el total de usuarios que contribuyen con la promoción del gestor en el país mediante acciones de capacitación, desarrollo, fomento, entre otros.
- La gestión de eventos, no pudiendo desde este espacio inscribirse en los eventos gestionados por el GUP.cu, subir ponencias para su presentación, visualizar el listado de eventos ocurridos, eventos próximos, total de participantes, entre otros.
- La gestión de actividades de capacitación, no pudiendo desde este espacio realizar solicitudes de matrícula en estas actividades, visualizar el listado de actividades ocurridas, actividades próximas, total de participantes, entre otros.

La última versión del portal fue desarrollada sobre Joomla 3.0 y PostgreSQL 9.0 a finales del año 2012, por lo que actualmente, el portal en línea presenta vulnerabilidades, alertadas por la Dirección de Redes y Seguridad Informática de la Universidad, donde se publica de cara a Internet.

### **1.2.2. Portales web del Grupo de Usuarios PostgreSQL de Argentina y de la Comunidad Brasileña de PostgreSQL**

Desde sus portales, disponibles en <https://www.arpug.com.ar.xthexder.info> y <https://www.postgresql.org.br>, estas comunidades publican noticias, documentación e instaladores de herramientas para el trabajo con el gestor (controladores ODBC para *Windows*, herramientas administrativas, entre otras), instaladores del gestor y un listado de eventos, pero: (ArPUG, 2008), (Comunidad de PostgreSQL Brasileña, 2015)

- No se puede constatar la gestión de los miembros del Grupo.
- No permiten la gestión de eventos, solamente muestra un listado con información de cada uno.
- No permiten la gestión de actividades de capacitación.

### **1.2.3. Portal web en español sobre PostgreSQL**

Sitio dedicado a la promoción del gestor en idioma español sin estar asociado a una comunidad de un país en particular, sino a la comunidad hispanohablante, disponible en <http://www.postgresql.org.es/comunidad>; muestra información detallada y actualizada, permite el registro de nuevos miembros, cuenta con una sección de encuestas para conocer los criterios de sus lectores y miembros, cuenta con un foro donde se debate sobre temas relacionados con PostgreSQL, cuenta con una sección de respuestas a preguntas de soporte, servicio y muchas otras (PostgreSQL-es, 2013).

Al no estar asociado a una comunidad en particular:

- No gestiona miembros.
- No gestiona eventos.
- No gestiona actividades de capacitación.

En la tabla 1 se muestra, un análisis de los portales de las principales comunidades de Hispanoamérica, en función de los requerimientos que debe poseer el Grupo de Usuarios PostgreSQL de Cuba.

Tabla 1. Comparación de los portales de las comunidades de Hispanoamérica.

Elementos	Portales de PostgreSQL			
	Cuba	Argentina	Brasil	Hispano-hablante
Información actualizada	No	No	No	Sí
Descarga de documentación	Sí	Sí	Sí	Sí
Descarga de instaladores	Sí	Sí	Sí	Sí
Gestión de miembros	No	No	No	No
Gestión de eventos	No	No	No	No
Gestión de actividades de capacitación	No	No	No	No

Una vez realizado el estudio de las características fundamentales con que cuentan los principales portales de la región teniendo en cuenta las necesidades actuales del GUP.cu, se concluye que ninguno de los portales analizados solventa en su totalidad dichas necesidades, además de que no pueden ser tomados y personalizados; por tanto, se hace necesario el desarrollo de una nueva solución que dé respuesta a los requerimientos del GUP.cu, teniendo en cuenta que el actual está desarrollado sobre tecnologías, si bien no obsoletas, ya en desuso, generando problemas de seguridad y de soporte futuro.

### 1.3. Metodología, tecnologías y herramientas para el desarrollo de la solución

Para el desarrollo de la solución se hace necesario el empleo de una metodología que guíe el proceso de desarrollo de software, así como un grupo de tecnologías y herramientas necesarias para su implementación.

#### 1.3.1. Metodología de desarrollo de software

Una metodología de desarrollo de software se refiere al entorno que se usa para estructurar, planificar y controlar el proceso de desarrollo de un sistema de información (Romero, 2012). Para guiar el proceso de desarrollo de la solución propuesta se escogió la variación de la Metodología Unificada Ágil (AUP) definida por la UCI.

La metodología unificada ágil (AUP) es una forma simplificada del Proceso Racional Unificado (RUP, *Rational Unified Process*) desarrollada por Scott Ambler. Describe un enfoque simple del desarrollo del software usando técnicas y conceptos ágiles. Algunas técnicas usadas por AUP incluyen el desarrollo orientado a pruebas, modelado y gestión de cambios ágiles y refactorización de base de datos para mejorar la productividad (Selva, 2014).



De las 4 fases que propone AUP (Inicio, Elaboración, Construcción, Transición), AUP-UCI mantiene la fase de Inicio, unifica las 3 fases restantes en una sola llamada Ejecución y agrega la fase de Cierre: (Sánchez, 2015)

- Inicio: en esta fase se realiza un estudio inicial de la organización cliente que permite obtener información fundamental acerca del alcance del proyecto, realizar estimaciones de tiempo, esfuerzo y costo y decidir si se ejecuta o no el proyecto.
- Ejecución: en esta fase se ejecutan las actividades requeridas para desarrollar el software, incluyendo el ajuste de los planes del proyecto considerando los requisitos y la arquitectura; durante el desarrollo se modela el negocio, se obtienen los requisitos, se elaboran la arquitectura y el diseño, se implementa y se libera el producto.
- Cierre: en esta fase se analizan tanto los resultados del proyecto como su ejecución y se realizan las actividades formales de cierre del proyecto.

### **1.3.2. Sistemas de Gestión de Contenidos**

Un Sistema de Gestión de Contenidos (*Content Management System*, en inglés, abreviado CMS) permite la creación y administración de contenidos principalmente en páginas web. Consiste en una interfaz que controla una o varias bases de datos donde se aloja el contenido del sitio. El sistema permite manejar, de manera independiente, el contenido y el diseño (Cuerda, 2004).

Entre las ventajas de los CMS destacan que permiten: (Domínguez, y otros, 2006)

- Manejar eficientemente gran cantidad de páginas web.
- La inmediatez de la publicación y edición de los contenidos.
- Trabajar en un ambiente de páginas web interactivas, generadas según las peticiones de los usuarios.
- Controlar el acceso de los usuarios al sistema, no solo mediante su contraseña, sino mediante los permisos asignados a cada.
- La inclusión de nuevas funcionalidades en la web, mediante la adición de módulos realizado por terceros, sin que implique muchos cambios en la web (Cuerda, 2004).
- La reutilización de objetos o componentes (Cuerda, 2004).

Actualmente existe una gran variedad de CMS; algunos de los más utilizados según la bibliografía consultada son *WordPress*, *Radiant CMS*, *SilverStripe*, *Joomla*, *TYPOlight*, *Frog*, *Textpattern*, *Drupal*, *ExpressionEngine*, y *CMS Made Simple* (WM, 2009).

Para el desarrollo de la solución requerida por el Grupo de Usuarios PostgreSQL de Cuba se realiza un análisis de los CMS *WordPress*, *Joomla* y *Drupal*, que tienen la particularidad de ser de código abierto, tienen una comunidad de desarrollo fuerte y cuentan con buena documentación (WM, 2009).

### **WordPress 4.6.1**

WordPress es una avanzada plataforma semántica de publicación personal, de código abierto, orientada a la estética, los estándares web y la usabilidad (wordpress, 2016). Se ha transformado de una plataforma básica de publicación de blogs a un CMS completamente funcional. Es intuitivo, brinda opciones para la creación de apariencia, publicaciones, páginas, entre otros. Además, está bien organizado y es fácil encontrar dónde desarrollar las diferentes funcionalidades (Martínez, y otros, 2012).

Como sistema de gestión de contenidos, WordPress consiste en un programa que permite una estructura de soporte para que usuarios de Internet puedan crear y administrar contenidos. Un CMS controla bases de datos que permiten manejar el diseño y el contenido de una página web. Es por eso que con WordPress, los usuarios pueden cambiar, cuantas veces quieran, el diseño de su página o blog, sin necesidad de cambiar el formato. Además, permite la publicación de material de una manera sencilla para luego ser vista por otros usuarios (wordpress, 2016).

### **Joomla 3.6**

Joomla es actualmente considerado uno de los CMS de código abierto más populares. La interfaz es relativamente simple y directa, con distintas secciones para manejar artículos, bloques, lista de vínculos, multimedia y contenidos. Este CMS es bastante poderoso, por lo que no es sugerido para sitios simples, en los que se tendría un exceso de funcionalidades. Incluye un número de funcionalidades para hacer que las páginas carguen más rápido, incluyendo compresión cache y *gzip*; desde el punto de vista de optimización, una de sus desventajas es que utiliza tablas para el diseño, al contrario de otros CMS que utilizan la etiqueta *div* (Martínez, y otros, 2012).

Una de las ventajas por las cuales se destaca es que con este solo preocupa la información que se desea publicar ya que este CMS gestionará todos los detalles técnicos y administrativos. No obstante, hay varios aspectos negativos relacionados a Joomla como es el caso de la usabilidad, tiene además el problema de ser muy rígido en su organización del contenido, con una estructura jerárquica en dos niveles, esto obliga a planificar muy bien qué contenido tendrá el sitio y cómo se va a estructurar, porque no es sencillo cambiarlo una vez se tenga mucho publicado. Entre otras de sus limitantes más significativas resaltan su alta dependencia de *javascript* en su panel de administración y el cierto nivel de lentitud que presenta en su funcionamiento (Joomla, 2017).

## Drupal 7.42

Drupal es otro CMS de código abierto bastante poderoso que puede ser utilizado para el desarrollo desde sitios corporativos hasta sitios de comercio electrónico o redes sociales. La interfaz es simple, con vínculos lógicamente organizados para crear y/o editar nuevos contenidos o administrar usuarios y permisos; posee una gran variedad de módulos disponibles para añadir funcionalidades más allá del contenido básico de administración. Su código es un poco más complejo que el de otros CMS, pero no deja de ser relativamente semántico y descifrable. Posee una amplia comunidad de usuarios, con foros en el sitio principal de desarrollo y con gran cantidad de documentación, tanto para desarrolladores como para usuarios finales (Martínez, y otros, 2012).

Es un programa de código abierto, con licencia GNU/GPL<sup>1</sup>, escrito en PHP, desarrollado y mantenido por una activa comunidad de usuarios. Ofrece una gran consistencia de todo el sistema. Es un sistema dinámico, en lugar de almacenar los contenidos en archivos estáticos en el sistema de ficheros del servidor de forma fija, el contenido textual de las páginas y otras configuraciones son almacenados en una base de datos y se editan utilizando un entorno Web incluido en el producto (Drupal, 2017).

Dentro de sus características que lo hacen uno de los más utilizados en el desarrollo de aplicaciones se encuentran: (Drupal Hispano, 2005)

- Ayuda en línea: un robusto sistema de ayuda en línea y páginas de ayuda para los módulos del núcleo, tanto para usuarios como para administradores.
- Búsqueda: todo el contenido en Drupal es totalmente indexado en tiempo real y se puede consultar en cualquier momento.
- Código abierto: el código fuente de Drupal está libremente disponible bajo los términos de la licencia GNU/GPL. Al contrario de otros sistemas de blogs o de gestión de contenido propietarios, es posible extender o adaptar Drupal según las necesidades.
- Módulos: la comunidad de Drupal ha contribuido con muchos módulos que proporcionan funcionalidades como página de categorías, autenticación mediante jabber, mensajes privados.
- Personalización: un sólido entorno de personalización está implementado en el núcleo de Drupal. Tanto el contenido como la presentación pueden ser individualizados de acuerdo a las preferencias definidas por el usuario.

Además, Drupal cuenta con varias ventajas: (Drupal, 2017)

---

<sup>1</sup> GNU/GPL: Licencia Pública General de GNU o más conocida por su nombre en inglés GNU General Public License (o simplemente sus siglas del inglés GNU/GPL).

- Configuración de idiomas, traducción al español casi completamente, mucha información tanto en inglés como en español, gran comunidad de usuarios aportando información.
- Fácil configuración de temas y una gran cantidad de plantillas.
- Gran cantidad de opciones de configuración con una interfaz amigable para el administrador.
- Fácilmente extensible a través de módulos que permiten adaptar rápidamente la aplicación a necesidades específicas, con más de 540 módulos para extender.

Después del análisis de los CMS de código abierto ampliamente utilizados, se decide emplear Drupal en su versión 7.42, teniendo en cuenta todas las potencialidades que ofrece.

### **1.3.3. Herramienta y lenguaje para el modelado**

En el siguiente epígrafe se detallan las herramientas y el lenguaje para el modelado de la presente investigación.

#### **Visual Paradigm**

Las herramientas CASE (*Computer Aided Software Engineering*; Ingeniería de Software Asistida por Ordenador) se pueden definir como un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores durante todos los pasos del ciclo de vida de desarrollo de un software (Reyna, 2012).

Visual Paradigm es una plataforma diseñada para acelerar el proceso de análisis y diseño de aplicaciones empresariales complejas a través del Lenguaje Unificado de Modelado. Esta herramienta soporta el ciclo completo del desarrollo de software, permite diseñar todos los tipos de diagramas de clases, generar código desde diagramas y generar documentación (Cabrera, y otros, 2012).

Además, es la herramienta utilizada en la Universidad para fines académicos y de desarrollo de software, pudiendo emplearse legalmente el poseer la institución una licencia de desarrollo.

#### **Lenguaje Unificado de Modelado**

El Lenguaje Unificado de Modelado (conocido por sus siglas en inglés UML, *Unified Modeling Language*) es la sucesión de una serie de métodos de análisis y diseño orientadas a objetos. Es llamado un lenguaje de modelado, no un método; utilizado para especificar, visualizar, construir y documentar los elementos de los sistemas de software (González, 2008).

Permite una comunicación sencilla entre clientes y desarrolladores, ya que los clientes aun sin tener conocimientos del lenguaje pueden entender lo modelado por los desarrolladores.

### 1.3.4. Lenguajes de programación y de diseño

Un lenguaje de programación es un idioma artificial diseñado para expresar órdenes a ser interpretadas por las computadoras, diseñado para describir el conjunto de acciones consecutivas que un equipo debe ejecutar; siendo un modo práctico para que los seres humanos puedan dar instrucciones a una computadora (PostgreSQL.it, 2009).

#### Lenguaje de marcado: HTML 5

HTML, del acrónimo en inglés *HiperText Markup Language*, en español Lenguaje de Marcado de Hipertexto, es un lenguaje de programación muy sencillo que se utiliza para crear los textos y las páginas web. Esta definición se debe a que está compuesto por etiquetas que definen la estructura y el formato del documento que verá el usuario a través de la web. Esas etiquetas son leídas por el navegador o visualizador, es decir el programa que se utiliza para navegar, y que es el que ejecuta las funciones creadas en HTML permitiendo que puedan ser visibles en nuestra máquina (definición, 2017).

Además, el HTML permite incluir *scripts*, códigos que pueden modificar el comportamiento de los navegadores web y de otros procesadores de HTML. Dentro de las ventajas que presenta se pueden mencionar: (Lamarca, 2013)

- Texto presentado de forma estructurada y agradable.
- No necesita de grandes conocimientos cuando se cuenta con un editor de páginas web.
- Archivos pequeños.
- Proporciona una mayor optimización de la velocidad y un mejor uso del hardware.
- Agrega elementos como audio y video, que permiten incluir y controlar contenido multimedia en la web sin tener que recurrir a *plugins* y APIs propietarias.

#### Lenguaje de diseño gráfico: CSS 3

CSS, del inglés *Cascading Style Sheets*, traducido como Hojas de Estilo en Cascada, es un lenguaje que describe la presentación de los documentos estructurados en hojas de estilo para diferentes métodos de interpretación, especificando cómo se va a mostrar un documento en pantalla (W3C, 2017).

CSS 3 es una especificación desarrollada por el W3C (*World Wide Web Consortium*), traducido como Consorcio Mundial de la Red, para permitir la separación de los contenidos de los documentos escritos en HTML, XML, XHTML, SVG, o XUL de la presentación del documento con las hojas de estilo, incluyendo elementos tales como los colores, fondos, márgenes, bordes, tipos de letra, modificando la apariencia de páginas web de forma sencilla, permitiendo a los desarrolladores controlar el estilo y formato de sus documentos (W3C, 2017).

Su uso permite: (W3C, 2017)

- Aumentar la accesibilidad de los usuarios, gracias a que permite especificar su propia hoja de estilo permitiéndoles modificar el formato de un portal según sus necesidades.
- Lograr una apariencia uniforme en toda la aplicación.
- Hacer que el código HTML sea más fácil de leer.
- Permitir que las páginas carguen más rápido.

### **Lenguaje de programación del lado del cliente: JavaScript 1.8.5**

JavaScript es un lenguaje de programación utilizado para crear pequeños programas encargados de realizar acciones dentro del ámbito de un portal web. Es un lenguaje del lado del cliente porque es el navegador el que soporta la carga de procesamiento. Su uso se basa fundamentalmente en la creación de efectos en el diseño de las páginas y la definición de interactividades con el usuario (JS, 2016).

Es un lenguaje con muchas posibilidades, permite la programación de *scripts*, pero también de programas grandes, orientados a objetos, con funciones, estructuras de datos complejas, entre otros (JS, 2016).

Entre las características que más utilidades pueden aportar al sistema que se desarrolle con este lenguaje se encuentran: (desarrolloweb, 2012)

- Rapidez en la ejecución del código de la aplicación a desarrollar, ya que este lenguaje es interpretado por el navegador que recibe el programa, no se compila.
- No se declaran los tipos de variables, por lo que se ahorra tiempo en la implementación de algunas funcionalidades del sistema.
- Permite la ejecución dinámica, respondiendo a eventos producidos por el usuario en tiempo real.

Drupal ha incorporado mucho JavaScript (por ejemplo, la librería jQuery) a su código fuente, esto no afecta la usabilidad de una aplicación basada en este CMS, al contrario, solo aumenta y optimiza su funcionalidad.

### **Lenguaje de programación del lado del servidor: PHP 5.6.2**

Para el desarrollo de la solución se hace uso del lenguaje de programación PHP, es un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor. La meta del lenguaje es permitir a los desarrolladores la generación dinámica de páginas. Con PHP se puede realizar el procesamiento de información en formularios, foros de discusión y páginas dinámicas. Además, permite la integración con varias bibliotecas externas, brindándole al desarrollador mayores opciones, desde generar documentos en PDF hasta analizar código XML (Lenguaje de Marcas Extensible, siglas del inglés *Extensible Markup Language*) (php, 2001).

PHP ofrece una solución simple y universal para las paginaciones dinámicas de la web de fácil programación. Su diseño elegante lo hace, perceptiblemente, más fácil de mantener y ponerse al día en comparación con el código de otros lenguajes. Debido a su amplia distribución es soportado por una gran comunidad de desarrolladores. Como producto de código abierto, goza de la ayuda de un gran grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y se reparen rápidamente (php, 2001).

El código de PHP se actualiza continuamente con mejoras y extensiones para ampliar las capacidades del lenguaje. Está compuesto de un sistema principal, un conjunto de módulos y una variedad de extensiones de código. Posee interfaces distintas para cada tipo de servidor (Programacion, 2016).

PHP cuenta con un gran número de ventajas que lo hacen ser uno de los lenguajes de programación más utilizados, hoy día, para el desarrollo de páginas web, entre las que destacan que: (Programando la web, 2016)

- Es fácil de aprender.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Soporta la orientación a objeto, clases y herencia.
- Es multiplataforma, se puede ejecutar en plataformas Linux, Windows, entre otros.
- Ofrece capacidad de conexión con la mayoría de los sistemas de gestión de bases de datos como PostgreSQL, MySQL, Oracle, MS SQL Server, entre otras.
- Tiene la capacidad de expandir su potencial utilizando módulos.
- Posee documentación que incluye descripción y ejemplos de cada una de sus funciones.
- Incluye gran cantidad de funciones.

### **1.3.5. Sistema de gestión de bases de datos**

Un sistema de gestión de bases de datos (SGBD) es el software que permite a los usuarios procesar, describir, administrar y recuperar los datos almacenados en una base de datos (Date, 1999).

Es el software que permite a los usuarios procesar, describir, administrar y recuperar datos almacenados en una base de datos (Nevado, 2010).

Por todas sus ventajas y, además, que posee una licencia totalmente libre, se selecciona como gestor de bases de datos para el desarrollo del Portal para la gestión del Grupo de Usuarios PostgreSQL de Cuba.

## PostgreSQL 9.6

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD<sup>2</sup>, considerado uno de los sistemas de gestión de bases de datos de código abierto más potentes, al soportar gran parte del estándar SQL, tipos de datos, consultas complejas, disparadores y actualización de vistas y, poder ser extendido mediante la adición de tipos de datos, funciones, operadores, índices, lenguajes procedurales, entre otros. (PostgreSQL, 2017).

Entre sus principales ventajas destacan que: (Nipas, 2014)

- Permite la replicación asincrónica/sincrónica.
- Permite realizar copias de seguridad en caliente.
- Está licenciado bajo la licencia BSD, de código abierto.
- Está disponible para Linux, UNIX en todas sus variantes y Windows 32/64bit.
- Tiene una documentación completa.

### 1.3.6. Servidor web

Un servidor web es un programa que se ejecuta continuamente en una computadora, manteniéndose a la espera de peticiones de ejecución que le hará un cliente o un usuario de Internet; el servidor web se encarga de contestar a estas peticiones de forma adecuada, entregando como resultado una página web o información de todo tipo de acuerdo a los comandos solicitados (ALEGSA, 2016).

Dentro de los servidores web más usados, según (JLAYA, 2016) se encuentran Apache, *Microsoft IIS*, *Sun*, *nginx* y Google.

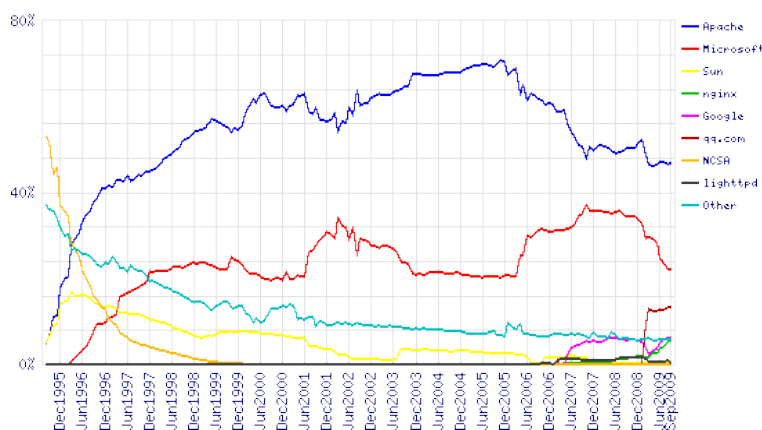


Figura 1. Gráfica de principales Servidores Web.

<sup>2</sup> Berkeley Software Distribution. Licencia de software libre permisiva.



En la presente investigación se analizan los servidores Apache y *Microsoft Internet Information Sever* por ser los más usados en el desarrollo web.

#### **Apache 2.4.4**

Apache es un servidor de licencia *freeware*<sup>3</sup> que está dominando el mundo informático por su amplio nivel de capacitación, su costo y su compatibilidad con los sistemas operativos (Apache, 2016).

Algunas características de Apache son que: (openSUSE, 2012)

- Puede ejecutarse en una gran variedad de sistemas operativos, lo que lo hace prácticamente universal.
- Es una tecnología gratuita de código fuente abierto.
- Es un servidor altamente configurable de diseño modular, por lo que es muy sencillo ampliar sus capacidades.
- Trabaja con los lenguajes Java, Perl y PHP, ofreciendo todo el soporte que se necesita para el trabajo con páginas dinámicas.
- Permite personalizar la respuesta ante los posibles errores que se puedan presentar en el servidor.
- Es altamente configurable en la creación y gestión de *logs*<sup>4</sup>; permite la creación de ficheros de *logs* necesarios para el administrador, ofreciendo mayor control sobre lo que sucede en el servidor.

Por sus características, Apache es un servidor web que se puede utilizar para la solución de la presente investigación.

#### **Microsoft Internet Information Server**

*Microsoft Internet Information Server*, mejor conocido como IIS, es la plataforma de servidores desarrollada por *Microsoft*, por lo que no es gratuita ni de código abierto, aunque la misma ha tenido un ascenso significativo debido a su compatibilidad con los *hosts*<sup>5</sup> y por sus nuevas innovaciones como .NET, *Silverlight*, entre otros. Uno de los factores más importantes sobre el ascenso de este servidor es que viene preinstalado en la mayoría de los sistemas operativos de *Microsoft* por lo cual fue el estándar de servidores por mucho tiempo (Microsoft, 2016).

---

<sup>3</sup> *Freeware*: programa que no requiere pago ni otra compensación.

<sup>4</sup> Grabación secuencial de registros en una base de datos o archivos.

<sup>5</sup> Un *host* o anfitrión es un ordenador que funciona como el punto de inicio y fin de las transferencias de datos, más comúnmente descrito como el lugar donde reside un sitio web.

IIS es un poco más intuitivo (cuadros de diálogo y ventanas) que Apache, pero a costa de una integración total en el sistema que hace difícil distinguir qué opciones afectan al servidor y cuáles al sistema operativo (Apache, 2016).

El servidor web que se elige para el desarrollo de la solución para el Grupo de Usuarios PostgreSQL de Cuba es Apache ya que posee una gran estabilidad, es confiable y altamente configurable. Además, es multiplataforma, se puede ejecutar en una amplia variedad de sistemas operativos. Por otra parte, es un servidor muy eficiente, y consume menos recursos del sistema en comparación a otros servidores. Se tuvo en cuenta que Apache es un software libre que se utiliza como servidor HTTP de código abierto, y a diferencia de IIS, es gratuito.

### **1.3.7. IDE de desarrollo NetBeans**

Las herramientas de desarrollo son programas que permiten a los desarrolladores crear software; soportan varios lenguajes de programación además de contar con interfaces, proporcionando un ambiente cómodo para el desarrollo de software (NetBeans, 2012).

NetBeans 8.2 es una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas; permite crear proyectos para web, escritorio o plataformas móviles; soporta lenguajes como Java, PHP, JavaScript, Ajax, Groovy y Grails y C++; cuenta con una amplia comunidad de desarrolladores y suficiente documentación; cuenta con un número importante de módulos para extenderlo y; es un producto libre y gratuito sin restricciones de uso (NetBeans, 2012).

Entre las ventajas que posee destacan que presenta una interfaz sencilla de usar, es multiplataforma y posibilita un rápido diseño de interfaz de desarrollo (Mendivelso, 2013).

## **Conclusiones del capítulo**

Una vez finalizado el presente capítulo se concluye que: el estudio de los portales de las principales comunidades del área hispanohablante, evidenció que ninguno de ellos realiza todos los procesos para la gestión de miembros, eventos y acciones de capacitación, características con las que debe contar el portal para el Grupo de Usuarios PostgreSQL de Cuba por lo que se decide implementar una nueva solución acorde a dichas solicitudes. El análisis de los CMS más empleados en la actualidad permitió definir como base para el desarrollo de la solución Drupal 7.42; guiado por la metodología de desarrollo de software AUP en su variación para la UCI; para el modelado de la solución se hace uso de Visual Paradigm y UML; como lenguajes de programación HTML 5, CSS 3, JavaScript 1.8.5 y PHP 5.6.2, como sistema de gestión de bases de datos PostgreSQL 9.6, como servidor web Apache 2.4.4 y como IDE de desarrollo NetBeans 8.2.

## CAPÍTULO 2: ANÁLISIS Y DISEÑO DEL PORTAL PARA EL GRUPO DE USUARIOS POSTGRESQL DE CUBA

En el capítulo se definen las características con las que debe contar el portal para el Grupo de Usuarios PostgreSQL de Cuba. Se realiza un modelo de dominio con el objetivo de lograr un mayor entendimiento de los principales conceptos asociados al sistema. Además, para su implementación, se realiza una descripción detallada de los requisitos y casos de uso definidos.

### 2.1. Modelo de dominio

Debido a que las fronteras del negocio, así como los actores y trabajadores no están bien definidos, se diseñó un modelo de dominio, en el que se explican los conceptos más significativos asociados al dominio del problema; su esencia es representar elementos del mundo real, no componentes del software; muestra conceptos, asociaciones entre estos y sus atributos con el objetivo de que los desarrolladores conozcan los términos significativos y las relaciones que existen entre ellos (Larman, 2004).

El portal del Grupo de Usuarios PostgreSQL de Cuba fue desarrollado para dar soporte a los procesos llevados a cabo por el GUP.cu, facilitando la comunicación entre sus miembros; actualmente permite la gestión de noticias, documentación e instaladores de herramientas.

El diagrama de clases del dominio de los procesos llevados a cabo por el Grupo de Usuarios PostgreSQL de Cuba se muestra en la siguiente figura.

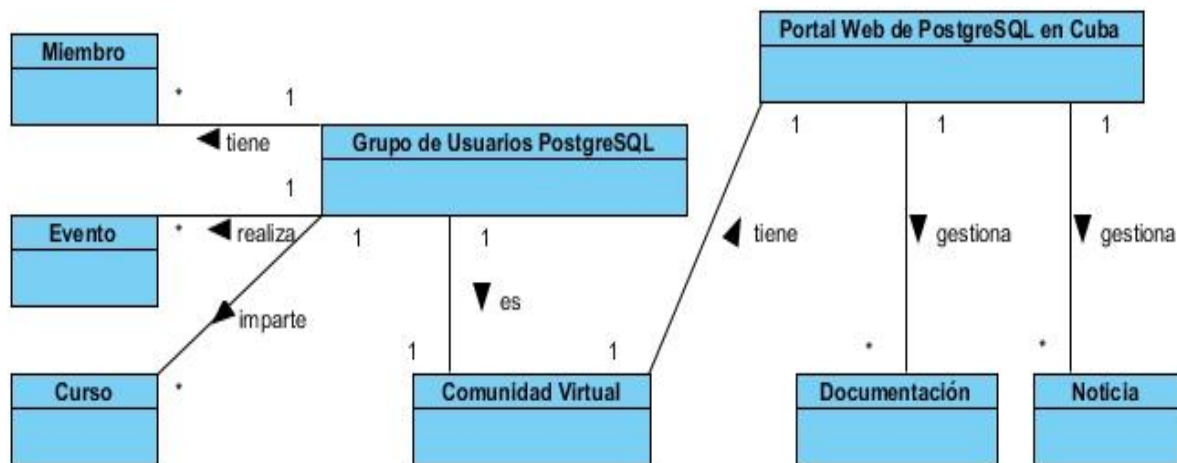


Figura 2. Diagrama de clases del dominio de los procesos del GUP.cu.

Los conceptos asociados se muestran en la tabla siguiente.

Tabla 2. Conceptos del diagrama de clases del dominio de los procesos del GUP.cu.

Concepto	Descripción
Grupo de Usuarios PostgreSQL	Comunidad de usuarios del país que desarrollan actividades (capacitación, divulgación, investigación, desarrollo) para la promoción del gestor
Comunidad virtual	Grupo de personas con objetivos comunes, que utilizan una misma forma de comunicación para interactuar entre ellos, en este caso mediante las Tecnologías de la Información y las Comunicaciones
Portal web	Herramienta que permite la gestión de noticias, documentación y la comunicación, de forma general, entre sus usuarios
Noticia	Novedad relacionada con el gestor
Documentación	Libros, guías, manuales y documentos, de manera general, que mediante su consulta posibilitan el estudio de determinado contenido, en este caso, relacionado con el gestor
Miembro	Persona perteneciente al Grupo de Usuarios PostgreSQL de Cuba, que colabora con el desarrollo de sus actividades
Evento	Actividad desarrollada por el Grupo de Usuarios PostgreSQL de Cuba con el propósito de promover el empleo del gestor en el país
Curso	Actividad de capacitación desarrollada por el Grupo de Usuarios PostgreSQL de Cuba con el propósito de que los usuarios que trabajen con el gestor estén preparados

## 2.2. Requisitos funcionales

Los requerimientos funcionales son los que se encargan de definir lo que la herramienta de software debe hacer. Definen los alcances del sistema en cuanto a las acciones que debe de realizar, y en cuanto a la transferencia de datos entre todas las diferentes funciones del sistema (Pressman, 2010).

En el caso del portal para el Grupo de Usuarios PostgreSQL de Cuba debe cumplir con los siguientes requisitos funcionales:

- RF 1. Registrar nuevo usuario
- RF 2. Modificar datos del usuario
- RF 3. Eliminar usuario existente
- RF 4. Mostrar usuarios existentes
- RF 5. Buscar usuarios
- RF 6. Autenticar usuario
- RF 7. Insertar nueva noticia
- RF 8. Modificar noticia

- RF 9. Eliminar noticia
- RF 10. Visualizar noticias existentes
- RF 11. Buscar noticias
- RF 12. Insertar nueva documentación
- RF 13. Modificar documentación
- RF 14. Eliminar documentación
- RF 15. Mostrar documentaciones existentes
- RF 16. Buscar documentación
- RF 17. Insertar nuevo instalador de herramienta
- RF 18. Modificar instalador de herramienta
- RF 19. Eliminar instalador de herramienta
- RF 20. Mostrar instaladores de herramientas existentes
- RF 21. Buscar instalador de herramienta
- RF 22. Insertar nuevo evento
- RF 23. Modificar datos del evento
- RF 24. Eliminar evento
- RF 25. Mostrar eventos registrados
- RF 26. Mostrar eventos realizados
- RF 27. Mostrar próximos eventos
- RF 28. Buscar eventos por nombre
- RF 29. Aprobar participante en evento disponible
- RF 30. Eliminar participante en evento
- RF 31. Mostrar participantes registrados en un evento
- RF 32. Subir ponencia a evento
- RF 33. Mostrar listado de ponencias enviadas a un evento
- RF 34. Insertar nueva acción de capacitación
- RF 35. Modificar acción de capacitación
- RF 36. Eliminar acción de capacitación
- RF 37. Mostrar acciones de capacitación registradas
- RF 38. Mostrar acciones de capacitación realizadas
- RF 39. Mostrar próximas acciones de capacitación
- RF 40. Buscar acciones de capacitación
- RF 41. Matricular en acción de capacitación
- RF 42. Crear tema en el foro de discusión

- RF 43. Modificar tema de foro de discusión
- RF 44. Eliminar tema de foro de discusión
- RF 45. Listar temas de foro de discusión
- RF 46. Publicar comentarios en foro de discusión
- RF 47. Descargar documentación e instaladores
- RF 48. Solicitar inscripción en evento
- RF 49. Listar solicitudes de acción de capacitación
- RF 50. Aprobar participante en acción de capacitación

### **2.3. Requisitos no funcionales**

Los requisitos no funcionales son las restricciones de los servicios o funciones ofrecidas por el sistema, restringen el espacio de posibles soluciones (Sommerville, 2005).

#### **Requerimientos de software**

Se necesitan como requerimientos mínimos:

- Para la PC cliente: navegador web Mozilla Firefox 35.
- Para el servidor de la aplicación: sistema operativo Windows Server, Linux (cualquier distribución), servidor web: Apache 2.4.4.
- Para el servidor de BD: PostgreSQL 9.6.

#### **Requerimientos de hardware**

Se necesitan como requerimientos mínimos:

- Para el cliente: procesador Intel Pentium IV, 512 MB de RAM, 256 GB de disco duro.
- Para el servidor: procesador Intel Core i5 a 3.0 GHz, 8 GB de RAM, 1 TB de disco duro.

#### **Requerimientos de restricción de diseño e instalación**

- Lenguaje de programación: PHP 5.6.2.

#### **Requerimientos de rendimiento**

- El tiempo de respuesta no debe exceder de los 5 segundos.
- El sistema debe soportar hasta 100 usuarios conectados concurrentemente sin afectar su rendimiento.

## 2.4. Análisis del sistema

En esta etapa se identifican los actores y casos de uso que tendrá el portal para el Grupo de Usuarios PostgreSQL de Cuba.

### 2.4.1. Actores del sistema

Un actor es quien interactúa con el sistema, inicializa los casos de uso, este puede ser un humano, una máquina o un software. Para el desarrollo de la solución se definen los siguientes actores:

- Administrador del sistema: rol que representa una persona que tendrá todos los privilegios para gestionar miembros del GUP.cu, noticias, documentación, instaladores, eventos, acciones de capacitación y foros.
- Usuario: rol que representa a cualquier persona no registrada en el portal que podrá leer noticias, información de eventos, acciones de capacitación y actividades en los foros, descargar documentación e instaladores.
- Miembro del GUP.cu: rol que representa a cualquier persona previamente registrada en el sistema, que podrá inscribirse como participante en los eventos, así como solicitar matrícula en las acciones de capacitación existentes.

### 2.4.2. Casos de uso del sistema

La tabla siguiente muestra los casos de uso identificados a partir de los requisitos funcionales pactados con el cliente.

Tabla 3. Casos de uso del sistema.

Referencia al requisito	Nombre del caso de uso	Prioridad
RF1	Registrar usuario	Crítico
RF2, RF3, RF4, RF5	Gestionar usuario	Secundario
RF6	Autenticar usuario	Secundario
RF7, RF8, RF9, RF11	Gestionar noticia	Crítico
RF10	Visualizar noticia	Crítico
RF12, RF13, RF14, RF15, RF16	Gestionar documentación	Crítico
RF17, RF18, RF19, RF20, RF21	Gestionar instalador herramienta	Crítico
RF22, RF23, RF24, RF28	Gestionar evento	Crítico
RF25, RF26, RF27	Visualizar evento	Crítico
RF29, RF30, RF31	Administrar participante	Crítico

RF32	Subir ponencia	Crítico
RF33	Mostrar listado de ponencia	
RF34, RF35, RF36, RF40	Gestionar acción capacitación	Crítico
RF37, RF38, RF39	Visualizar acción capacitación	Crítico
RF41	Matricular acción de capacitación	Crítico
RF42, RF43, RF44, RF45	Gestionar temas de foro de discusión	Crítico
RF46	Publicar en el foro de discusión	Crítico
RF47	Descargar documentación e instaladores	Crítico
RF48	Solicitar inscripción en evento	Crítico
RF49, RF50	Aprobar participante en acción de capacitación	Crítico

### 2.4.3. Diagrama de casos de uso del sistema

El diagrama de casos de uso refleja cómo los actores interactúan con los casos de usos; la figura siguiente muestra el diagrama correspondiente al portal para la gestión del Grupo de Usuarios PostgreSQL de Cuba.

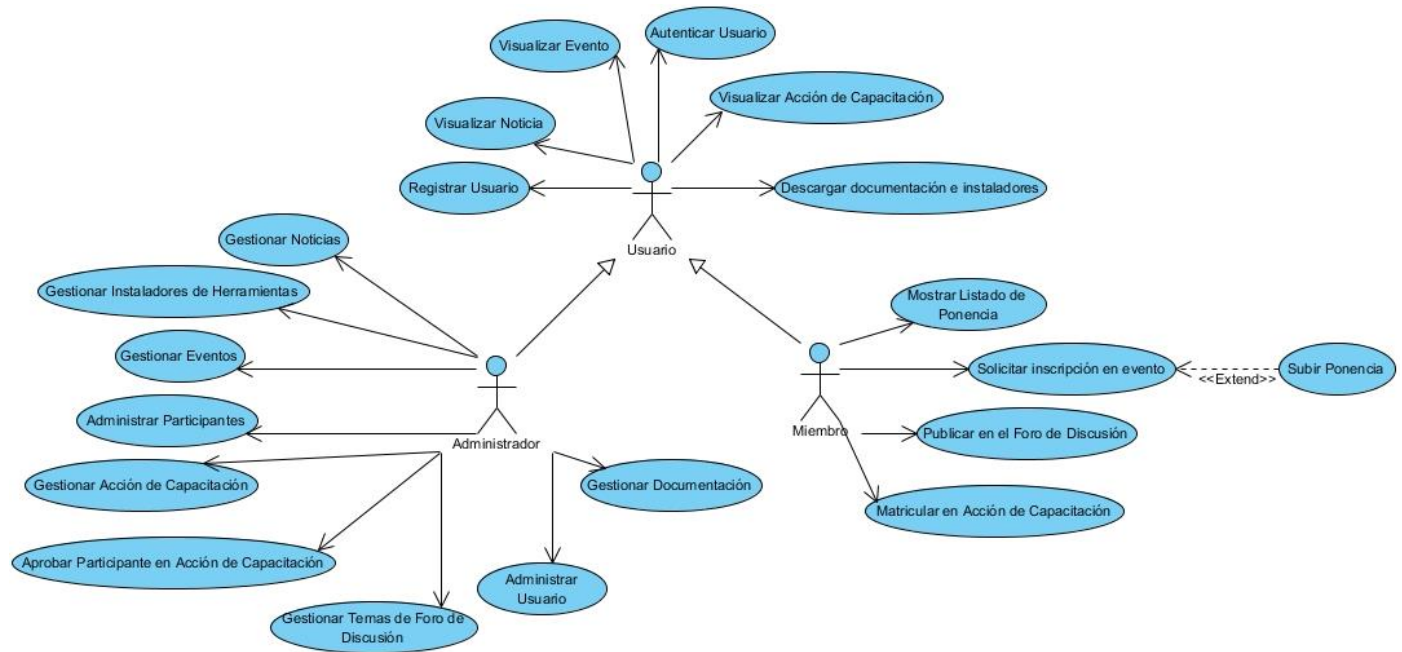


Figura 3. Diagrama de casos de uso del sistema.

### Patrones de caso de uso

A continuación, se presentan los patrones empleados en la realización del diagrama de casos de uso del sistema:



- **CRUD total:** permite modelar las diferentes operaciones para gestionar una entidad de información, tales como crear, mostrar, modificar y eliminar; ejemplo de aplicación de este es el CU Gestionar Evento, que lo inicializa el Administrador, permitiéndole crear, modificar, mostrar y eliminar un evento.
- **CRUD parcial:** permite modelar las diferentes operaciones para administrar una entidad de información, tales como insertar, eliminar y modificar; ejemplo de aplicación de este es el CU Administrar Participante, el cual lo inicializa el Administrador, permitiéndole aprobar, eliminar y modificar un participante en un evento.
- **Múltiples actores rol común:** permite representar cuando a un caso de uso ingresan más de dos actores con un rol común; este patrón se evidencia en los actores Administrador y Miembro que heredan del actor Usuario.
- **Extensión abstracta:** Consiste en dos casos de uso y una relación extendida entre ellos. Puede ser instalado en sí mismo, así como extendido en el caso de uso base. Este patrón se aplica cuando de un flujo puede extender el flujo de otro caso de uso, así como ser realizado en sí mismo; este patrón se evidencia en el CU Solicitar inscripción en evento que de él extiende el CU Subir Ponencia, ya que se puede participar en un evento sin tener que subir una ponencia, en este caso se participa en forma de observador.

## Descripción de los casos de usos del sistema

Los casos de uso se elaboran adicionalmente para proporcionar considerablemente más detalles sobre la interacción. Los casos de uso se escriben a menudo informalmente (Pressman, 2010). Sin embargo, la descripción formal del CU Gestionar Eventos se muestra en la siguiente tabla para garantizar que se aborden todas las cuestiones claves.

Tabla 4. Descripción textual del caso de uso Gestionar Eventos.

<b>Objetivo</b>	Manipular los eventos
<b>Actores</b>	Administrador (inicializa)
<b>Resumen</b>	El caso de uso le permite al administrador insertar, eliminar, modificar, mostrar y buscar los eventos, finalizando el caso de uso cuando este no desea realizar otra acción
<b>Complejidad</b>	Alta
<b>Prioridad</b>	Crítico
<b>Precondiciones</b>	Que el actor esté autenticado en el sistema
<b>Postcondiciones</b>	En dependencia de lo seleccionado por el autor queda insertado, modificado, eliminado o se muestra un evento

<b>Flujo de eventos</b>		
<b>Flujo básico Gestionar Evento</b>		
	<b>Actor</b>	<b>Sistema</b>
1	Selecciona la opción: a) Insertar un evento. b) Modificar un evento. c) Mostrar un evento d) Buscar un evento e) Eliminar un evento.	
2		En dependencia de la opción seleccionada, ejecuta la sección: Sección 1: Insertar un evento. Sección 2: Modificar un evento. Sección 3: Mostrar un evento. Sección 4: Buscar un evento. Sección 5: Eliminar un evento.
3		Finaliza el caso de uso.
<b>Sección 1: "Insertar un evento."</b>		
<b>Flujo básico Insertar correctamente</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Selecciona la opción Agregar Contenido y escoge el tipo de contenido "Eventos".	
2.		Muestra una interfaz para insertar un evento.
3.	Introduce los datos del evento (Nombre, descripción, fecha, lugar e imagen).	
4.		Valida que los campos obligatorios no estén vacíos.
5.		Valida que el campo fecha sea un rango correcto.
6.		Inserta el evento y muestra el mensaje "Se insertó un nuevo evento."
<b>Prototipo de interfaz de usuario</b>		

Inicio>>Agregar contenido

**Nombre de Evento \***

**Detalles del Evento \***

**Fecha Evento \***

Mes  Día  Año

hasta

Mes  Día  Año

**Lugar**

**Imagen de Evento**

 Examinar...

Guardar Previsualizar

### Flujos alternos

#### Nº 4a Campo obligatorio vacío

	Actor	Sistema
4.1		Muestra un mensaje indicando que el campo vacío es obligatorio.

#### Nº 5a Rango de fecha incorrecto

	Actor	Sistema
5.1		Muestra un mensaje indicando que el rango de fecha es incorrecto.

### Sección 2: "Modificar un evento."

#### Flujo básico Modificar correctamente

	Actor	Sistema
1.		Ejecutar la sección Buscar Evento.
2.	Selecciona el evento que desea modificar.	
3.		Muestra los datos editables del evento seleccionada.
4.	Modifica los datos del evento.	
5.		Valida que los campos obligatorios no estén vacíos.
6.		Valida que el campo fecha sea un rango

		correcto.
7.		Modifica el evento y muestra el mensaje “Se modificó el evento.”
8.		Se actualiza el listado de eventos en la interfaz.

### Prototipo de interfaz de usuario

Inicio>>PGDay Cubano Edicion 2017

**Nombre de Evento \***

**Detalles del Evento \***

**Fecha Evento \***  
 Mes:  Día:  Año:

**hasta**  
 Mes:  Día:  Año:

**Lugar**

**Imagen de Evento**

### Flujos alternos

#### Nº 5a Campo obligatorio vacío

	Actor	Sistema
5.1		Muestra un mensaje indicando que el campo vacío es obligatorio.

#### Nº 6a Rango de fecha incorrecto

	Actor	Sistema
6.1		Muestra un mensaje indicando que el rango de fecha es incorrecto.

### Sección 3: “Mostrar un evento.”

#### Flujo básico Mostrar correctamente

	Actor	Sistema
1.		Ejecutar la sección Buscar Evento.

2.	Selecciona el evento que desea visualizar.	
3.		Muestra los datos del evento seleccionado.
4.	Cierra la interfaz.	

### Prototipo de interfaz de usuario

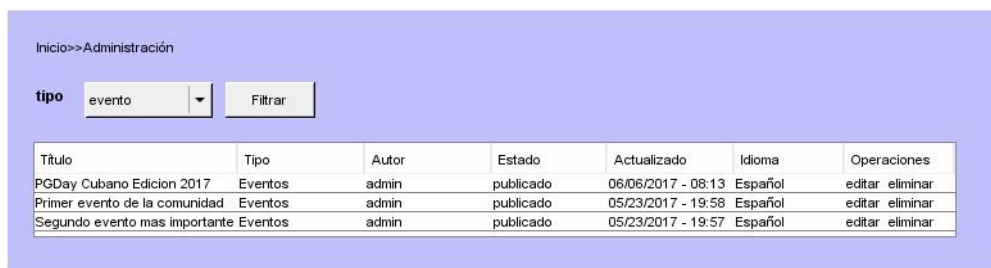


### Sección 4: “Buscar un evento.”

#### Flujo básico Buscar evento

	Actor	Sistema
1.		Muestra una interfaz para buscar un evento.
2.	Selecciona el tipo de contenido “eventos”.	
3.		Muestra un listado de todos los eventos.

### Prototipo de interfaz de usuario



### Sección 5: “Eliminar un evento.”

#### Flujo básico Eliminar correctamente

	Actor	Sistema
1.		Ejecuta la sección Buscar.
2.	Selecciona el evento que desea eliminar.	
3.		Muestra un mensaje “¿Desea eliminar el evento?”.
4.	Se confirma la eliminación.	
5.		Elimina el evento y muestra el mensaje “Se

		eliminó el evento.”
6.		Se actualiza el listado de evento en la interfaz.

### Prototipo de interfaz de usuario

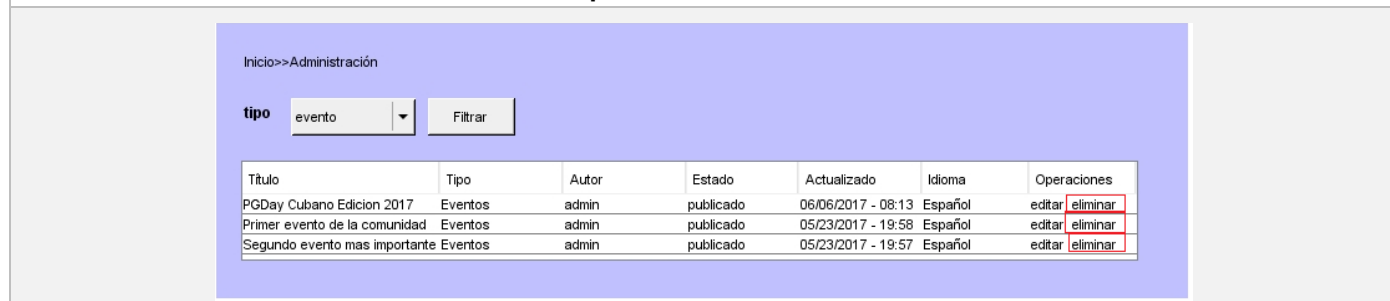


Tabla 5. Descripción textual del caso de uso Gestionar Acción de Capacitación.

<b>Objetivo</b>	Manipular las acciones de capacitación.
<b>Actores</b>	Administrador (inicializa)
<b>Resumen</b>	El caso de uso le permite al administrador insertar, eliminar, modificar y buscar las acciones de capacitación, finalizando el caso de uso cuando este no desea realizar otra acción.
<b>Complejidad</b>	Alta
<b>Prioridad</b>	Crítico
<b>Precondiciones</b>	Que el actor esté autenticado en el sistema.
<b>Postcondiciones</b>	En dependencia de lo seleccionado por el autor queda insertada, modificada, eliminada una acción de capacitación.

#### Flujo de eventos

#### Flujo básico Administrar Acción de Capacitación

	Actor	Sistema
1	Selecciona la opción: <ul style="list-style-type: none"> <li>a) Insertar una acción de capacitación.</li> <li>b) Modificar una acción de capacitación.</li> <li>c) Eliminar una acción de capacitación.</li> <li>d) Buscar una acción de capacitación.</li> </ul>	
2		En dependencia de la opción seleccionada, ejecuta la sección: Sección 1: Insertar una acción de capacitación. Sección 2: Modificar una acción de capacitación. Sección 3: Eliminar una acción de capacitación. Sección 3: Buscar una acción de capacitación.
3		Finaliza el caso de uso.

## Sección 1: "Insertar una acción de capacitación."

### Flujo básico Insertar correctamente

	Actor	Sistema
7.	Selecciona la opción Agregar Contenido y escoge el tipo de contenido "Curso".	
8.		Muestra una interfaz para insertar un curso.
9.	Introduce los datos del Curso (Título, Descripción, Profesor, Estado del Curso, Fecha, Matrícula).	
10.		Valida que los campos obligatorios no estén vacíos.
11.		Valida que el campo fecha sea un rango correcto.
12.		Inserta el curso y muestra el mensaje "Se insertó el curso."

### Prototipo de interfaz de usuario

Inicio>>Agregar contenido

**Título\***  
PGDay Cubano Edicion 2017

**Descripción\***  
Li European lingues es membres del sam familie. Lor separat existentie es un myth. Por scientie, musica, sport etc, litot Europa usa li sam vocabular. Li lingues differe solmen in li grammatica, li pronunciation e li plu commun vocabules. Omnicos directe al desirabilite de un nov lingua franca:

**Profesor\***  
Ing. Yandy Ernesto Cantillo

**Fecha del Curso\***  
Mes: 08, Día: 22, Año: 2017

**hasta**  
Mes: 08, Día: 22, Año: 2017

**Matrícula\***  
30

Guardar Previsualizar

<b>Flujos alternos</b>		
<b>Nº 4a Campo obligatorio vacío</b>		
	<b>Actor</b>	<b>Sistema</b>
4.1		Muestra un mensaje indicando que el campo vacío es obligatorio.
<b>Flujos alternos</b>		
<b>Nº 5a Rango de fecha incorrecto</b>		
	<b>Actor</b>	<b>Sistema</b>
5.1		Muestra un mensaje indicando que el rango de fecha es incorrecto.
<b>Sección 2: “Modificar una acción de capacitación.”</b>		
<b>Flujo básico Modificar correctamente</b>		
	<b>Actor</b>	<b>Sistema</b>
9.		Ejecutar la sección Buscar Evento.
10.	Selecciona la acción de capacitación que desea modificar.	
11.		Muestra los datos editables de la acción seleccionada.
12.	Modifica los datos de la acción a modificar.	
13.		Valida que los campos obligatorios no estén vacíos.
14.		Valida que el campo fecha sea un rango correcto.
15.		Modifica la acción de capacitación y muestra el mensaje “Se modificó la acción de capacitación.”
16.		Se actualiza el listado de acciones de capacitación en la interfaz.
<b>Prototipo de interfaz de usuario</b>		



Inicio>>Curso de PostgreSQL

**Título\***  
Curso de PostgreSQL

**Descripción\***  
Li European lingues es membres del sam familie. Lor separat existentie es un myth. Por scientie, musica, sport etc, litot Europa usa li sam vocabular. Li lingues differe solmen in li grammatica, li pronunciation e li plu commun vocabules. Omnicos directe al desirabilite de un nov lingua franca:

**Profesor\***  
Ing. Yandy Ernesto Cantillo

**Fecha del Curso\***  
Mes: 08, Día: 22, Año: 2017  
hasta  
Mes: 08, Día: 22, Año: 2017

**Matrícula\***  
30

Guardar      Previsualizar

### Flujos alternos

#### Nº 5a Campo obligatorio vacío

	Actor	Sistema
5.1		Muestra un mensaje indicando que el campo vacío es obligatorio.

### Flujos alternos

#### Nº 6a Rango de fecha incorrecto

	Actor	Sistema
6.1		Muestra un mensaje indicando que el rango de fecha es incorrecto.

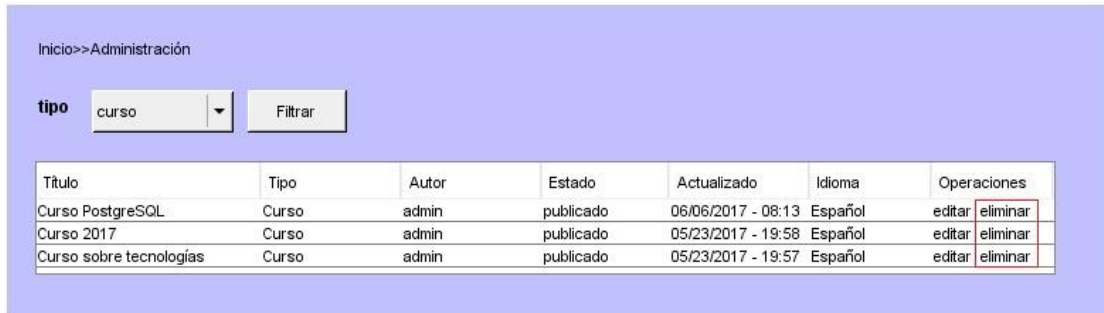
### Sección 3: “Eliminar una acción de capacitación.”

#### Flujo básico Eliminar correctamente

	Actor	Sistema
7.		Ejecuta la sección Buscar.
8.	Selecciona la acción de capacitación que desea eliminar.	
9.		Muestra un mensaje “¿Desea eliminar la

		acción de capacitación?”.
10.	Se confirma la eliminación.	
11.		Elimina la acción de capacitación y muestra el mensaje “Se eliminó la acción de capacitación.”
12.		Se actualiza el listado de acciones a capacitar en la interfaz.

### Prototipo de interfaz de usuario

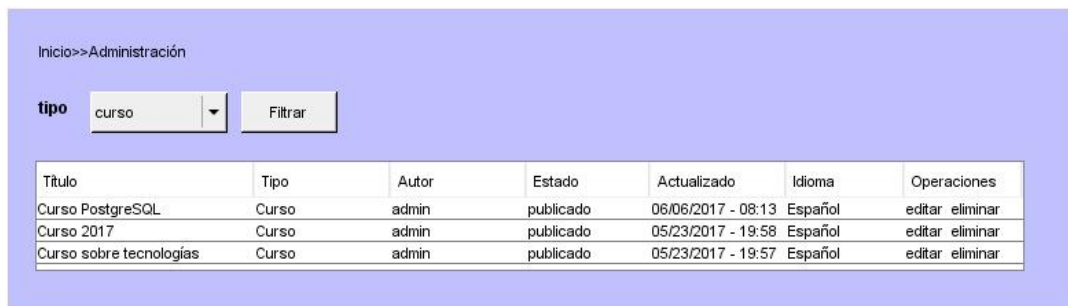


### Sección 4: “Buscar un acción de capacitación.”

#### Flujo básico Buscar correctamente

	Actor	Sistema
4.		Muestra una interfaz para buscar un curso.
5.	Selecciona el tipo de contenido “curso”.	
6.		Muestra un listado de todos los cursos.

### Prototipo de interfaz de usuario



<b>Relaciones</b>	<b>CU incluidos</b>	No procede
	<b>CU extendidos</b>	No procede

## 2.5. Diseño del portal para el Grupo de Usuarios PostgreSQL de Cuba

Según Pressman, el diseño del software es realmente un proceso de muchos pasos pero que se clasifican dentro de uno mismo; es donde se modela el sistema y su forma, en general, se refiere al establecimiento

de las estructuras de datos, la arquitectura general del software, representaciones de interfaz y algoritmos. El proceso de diseño traduce requisitos en una representación de software (Pressman, 2010).

El diseño de la arquitectura de software ocurre inmediatamente después de la especificación de los requerimientos de software y considera como elementos principales los componentes de software, las propiedades de dichos componentes y la comunicación entre ellos. El diseño detallado se lleva a cabo justo antes de la codificación y forma parte de las primeras tareas del desarrollador; describe la lógica, el control jerárquico, la estructura de datos, el empaqueo de componentes, entre otros (Pressman, 2010).

### **2.5.1. Arquitectura de software**

La arquitectura de software permite representar de forma concreta la estructura y funcionamiento interno de un sistema, lo que posibilita su entendimiento, la organización del desarrollo del mismo. La arquitectura de software de un sistema es la estructura o estructuras del sistema, lo cual abarca componentes de software, las propiedades visibles externamente de esos componentes y las relaciones entre ellas (Sommerville, 2005).

A continuación, se define la arquitectura para un mejor entendimiento y una organización del desarrollo del sistema.

El portal para el Grupo de Usuarios PostgreSQL de Cuba se implementa sobre el CMS Drupal, por lo tanto, la arquitectura y los patrones son heredados de este.

La particularidad de Drupal, para conseguir su reconocida flexibilidad y facilidad en la creación de sitios web, es la abstracción y organización en capas que aplica en el tratamiento de los contenidos. En lugar de considerar el sitio web como un conjunto de páginas interrelacionadas, Drupal estructura los contenidos en una serie de elementos básicos (nodos, módulos, bloques y menús, permisos de usuario y plantillas), ver figura 4, donde: (Drupal, 2017)

- Los nodos son los elementos básicos en que Drupal almacena la información, los contenidos.
- Los módulos son los elementos que operan sobre los nodos y otorgan funcionalidad a Drupal.
- Los bloques y menús permiten estructurar y organizar los contenidos en la página web.
- Con roles y permisos, Drupal dispone de un registro de usuarios y de roles que permiten especificar qué tareas pueden realizar y a qué contenidos puede acceder cada tipo de usuario.
- Las plantillas permiten establecer la apariencia gráfica o estilo de la información que se le muestra al usuario.

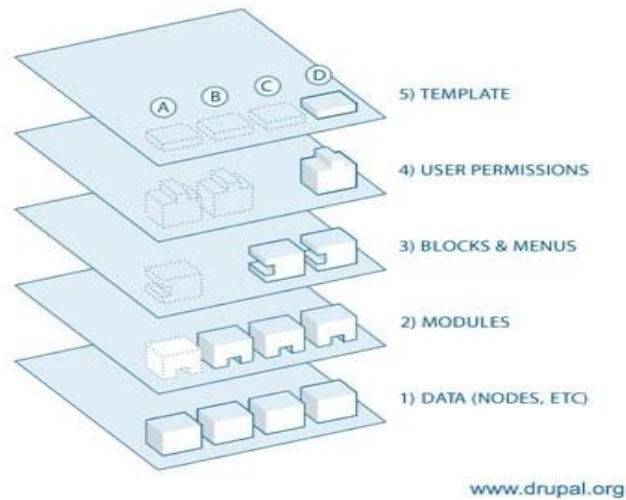


Figura 4. Estructura de Drupal.

Por tanto, Drupal es un sistema con una arquitectura modular que permite ampliar sus funcionalidades a través de unos métodos uniformes de desarrollo e integración de nuevos módulos. En última instancia un módulo consiste en un conjunto de archivos con código PHP que utiliza la arquitectura y las APIs de Drupal para incorporar nuevas características funcionales al sitio web (Gil, 2012).

El código que constituye el núcleo de Drupal está formado por un conjunto de librerías que permiten gestionar los procesos de arranque del sistema. Incluye además un conjunto de bibliotecas que brindan un grupo de servicios como conexión y administración de la base de datos, gestión de procesos de envío de correos, tratamiento de imágenes e internacionalización. En la Figura 5 se muestra, de forma esquemática, los elementos que conforman un sistema Drupal (Gil, 2012).

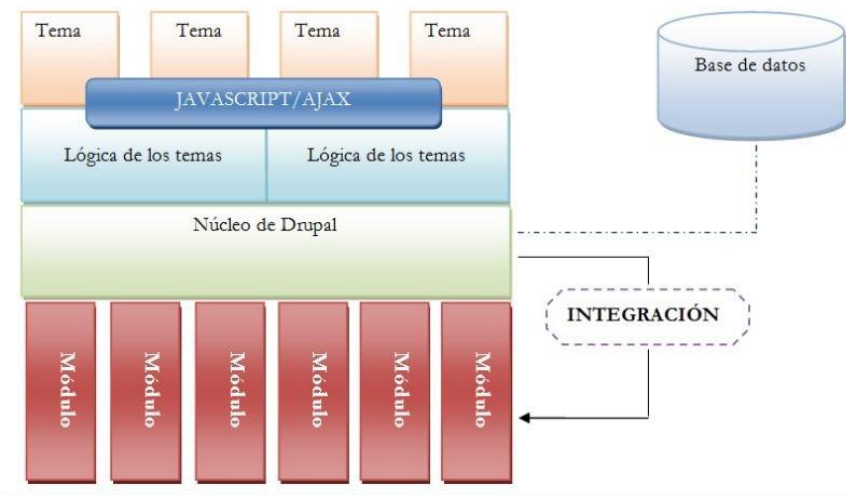


Figura 5. Arquitectura de Drupal.

## 2.6. Patrones

Un patrón es una descripción de un problema y su solución, que recibe un nombre y que puede emplearse en otros contextos; en teoría, indica la manera de utilizarlo en circunstancias diversas. Es un modo de proveer información en forma de una declaración de problema, algunas restricciones, una presentación de una solución ampliamente aceptada al problema, y luego una discusión de las consecuencias de esa solución. Muchos patrones ofrecen orientación sobre cómo asignar las responsabilidades a los objetos ante determinada categoría de problemas (Rosanigo, 2000).

### 2.6.1. Patrones de diseño

El núcleo de Drupal 7.42 está diseñado haciendo uso de una arquitectura modular. Las características estructurales presentes en este CMS son resultado de la aplicación de patrones de diseño.

A continuación, se describen cómo se aplican los patrones del diseño GoF<sup>6</sup> en la solución:

- *Singleton* (Instancia única): este patrón está diseñado para que solo permita que se instancie un elemento cuando este no existe; dentro del núcleo de Drupal se utiliza este patrón de diseño pensando en los módulos y temas como objetos para llevar a cabo la gestión de dichos elementos al solo crear una instancia de un nodo cuando este no existe. Este patrón se evidencia cuando se crea un tipo de contenido “Evento”, primero se verifica que no exista y luego se crea.
- *Command* (Comando): muchos de los *hooks* (ganchos) definidos por el núcleo de Drupal usan este patrón para reducir número de funciones que son necesarias implementar, pasando la operación como parámetro conjuntamente con los argumentos; de esta forma se evidencia en que cada módulo no necesariamente debe definir cada *hook*, sino solo los que tienen que ver con su propósito y funcionamiento. En el desarrollo de la solución propuesta se hace uso de este patrón ya que los módulos implementados no tienen que definir cada *hook*, sino los que vayan a aplicar, en la aplicación se hace uso de los módulos *Fontello*, *Date*, *Ctools*, *Views*, *Panels*, entre otros.
- *Bridge* (Puente): la capa de abstracción de datos de Drupal se encuentra implementada siguiendo el patrón *Bridge*; cada módulo debe ser programado de manera que sea independiente del motor de bases de datos que utiliza el sistema; lo que se logra por la capa de abstracción de bases de datos, sobre la que se pueden desarrollar nuevas capas, siguiendo la API definida, para añadir soporte para nuevos sistemas gestores de base de datos, sin necesidad de modificar el código de los módulos. Este patrón se evidencia ya que puede cambiarse la estructura del sitio o los tipos de

---

<sup>6</sup> *Gang of Four*, en español La Pandilla de los Cuatro.

contenidos sin que esto llegue a afectar un módulo en específico. Por ejemplo, si se elimina el tipo de contenido “Evento” no provoca que el módulo *Fontello* deje de funcionar en otro tipo de contenido donde este se emplee.

- *Observer* (Observador): los módulos que implementan un *hook* determinado por evento de inserción o actualización de una determinada entidad, son declarados como observadores de dichas entidades con las que interactúan. La utilización del patrón Observador se evidencia cuando se modifica un tipo de contenido que pertenece a una sección, entonces el sistema informa a todas sus dependencias de la modificación. Por ejemplo, si se modifica un contenido de tipo “Evento”, el sistema modifica este evento en los contenidos “Próximos eventos”, “Eventos Realizados” y a los “Eventos Registrados”.
- *Chain of responsibility* (Cadena de responsabilidades): el sistema de menús de Drupal es la evidencia de este patrón; en cada petición de una página, el sistema de menús determina si hay algún módulo para responder la petición, y qué función se debe llamar para procesar la petición; en este proceso se trasmite el mensaje de la petición por cada uno de los componentes que se encuentran inmersos; de esta forma se continúa la cadena hasta que un módulo atienda la petición, deniegue el acceso o hasta que la cadena se agote; todas las peticiones de rutas definidas por los módulos desarrollados evidencian este patrón. Un ejemplo de este patrón en el sistema es cuando se selecciona el vínculo “Búsqueda”, el sistema detecta el módulo de búsqueda que se encarga de esta petición, verifica si el usuario tiene permisos para usarlo y entonces le permite acceder a su contenido.

No se aplican patrones GRASP (*General Responsibility Assignment Software Patterns*, en español Patrones Generales de software para Asignación de Responsabilidades), ya que Drupal no implementa completamente los conceptos de la programación orientada a objetos.

## Conclusiones del capítulo

En el capítulo fueron descritas las características de la solución. Se diseñó un modelo de dominio para su mejor entendimiento; se identificaron 50 requisitos funcionales y 8 requisitos no funcionales, agrupados en 20 casos de uso del sistema, para los que se aplicaron los patrones de casos de uso CRUD total, CRUD parcial, múltiples actores rol común y extensión concreta; se definió como arquitectura de software la Modular, que es la empleada por Drupal, base de la solución propuesta y; fueron aplicados los patrones de diseño *Singleton*, *Command*, *Bridge*, *Observer* y *Chain of responsibility*.

## CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA

---

En el presente capítulo se presentan los elementos fundamentales para la implementación del sistema, el diagrama de despliegue y la vista de implementación como punto de partida para el flujo de trabajo de implementación. También se describen las pruebas a realizar definiéndose nivel, tipo de pruebas, métodos y técnicas a utilizar con el objetivo de validar y detectar posibles fallos en el portal para el Grupo de Usuarios PostgreSQL de Cuba.

### 3.1. Diagrama de despliegue

El diagrama de despliegue muestra la configuración de los nodos de proceso además de las instancias de objetos y componentes (Rumbaugh, y otros, 1999).

En la siguiente figura se muestra la distribución física necesaria para el despliegue del sistema. Para su correcto funcionamiento es necesario una PC cliente, que desde esta se podrá acceder al sitio, el cual estará alojado en un servidor web, que hace uso de un servidor de bases de datos para guardar toda la información necesaria que contenga el sitio.

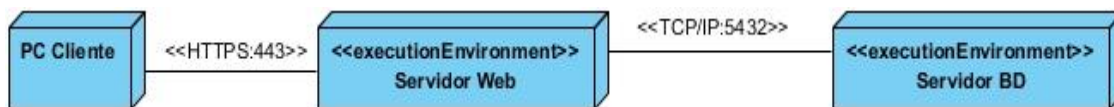


Figura 6. Diagrama de Despliegue del Grupo de Usuarios PostgreSQL de Cuba.

Donde:

- HTTPS: es el protocolo para establecer a través del puerto 443 la conexión segura entre el dispositivo de acceso cliente y el servidor de aplicaciones; la conexión es por cable vía modem, LAN o red inalámbrica con una velocidad de más de 64 Kbps.
- TCP/IP: es el protocolo para establecer la conexión entre el servidor de aplicaciones y el servidor de bases de datos; para el caso de PostgreSQL se utiliza el puerto 5432; la conexión entre el servidor web y el servidor de bases de datos permite dar órdenes y obtener información de este.

### 3.2. Vista de implementación

La vista de implementación contiene la organización de los módulos en términos de paquetes y capas, pueden incluirse también la trazabilidad de la vista lógica. Es representada por un diagrama de componentes o especificaciones de paquetes que son básicamente el subconjunto del modelo de despliegue. La vista de implementación constituye una selección de los aspectos fundamentales del diagrama de componentes del

sistema, el cual modela el empaquetado físico del sistema en unidades reutilizables llamadas “componentes” y sus relaciones (Mendoza, 2009).

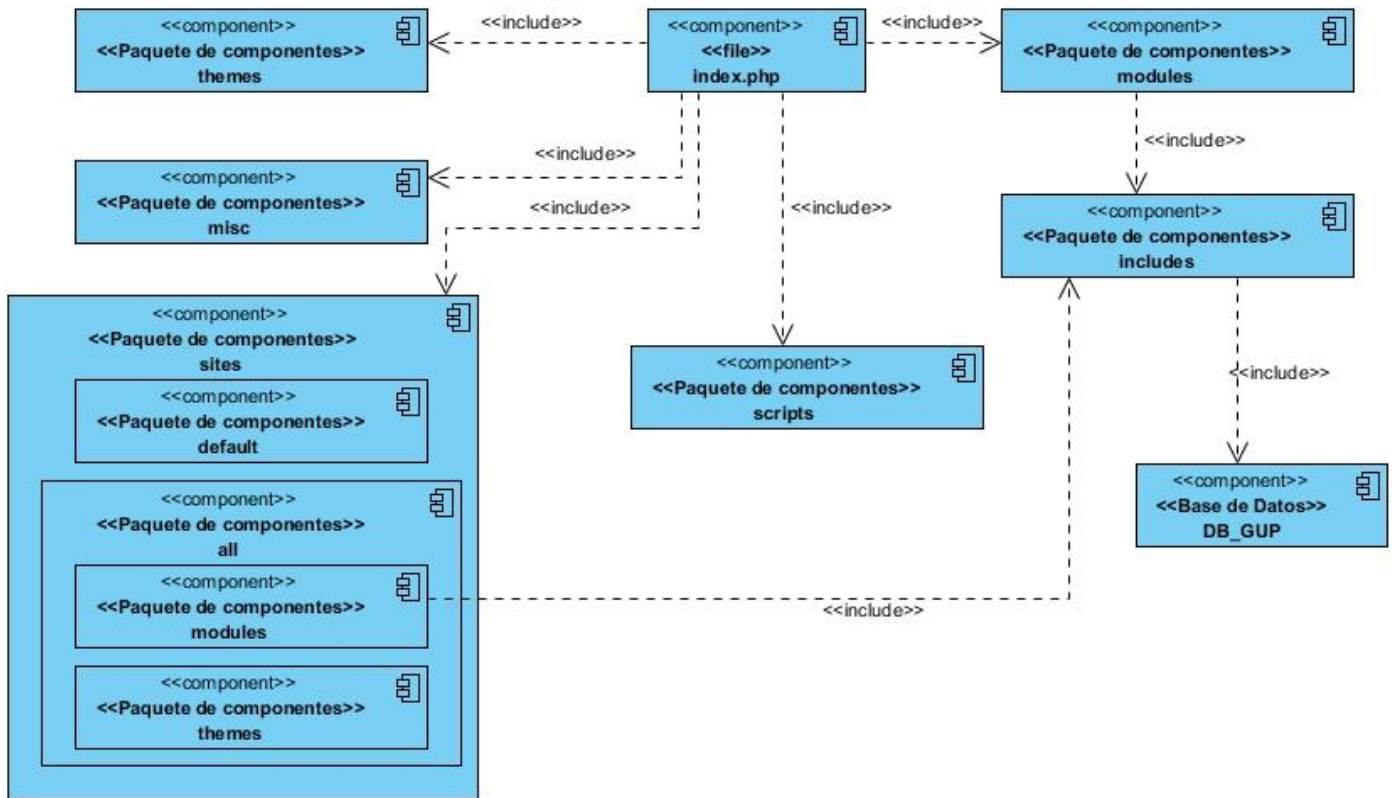


Figura 7. Vista de Implementación del GUP de Cuba.

En la siguiente tabla se muestran los elementos que componen la vista de implementación.

Tabla 6. Elementos que componen el Diagrama de Componentes.

Concepto	Descripción
index.php	Es la puerta de entrada al sistema. A partir de esta entrada se solicitan los diferentes módulos de Drupal. Cuando cargamos una página del sitio web estamos siempre haciendo una llamada a index.php. El procesamiento de la página comienza en el script index.php, que llama a la función drupal_bootstrap () y pone en marcha distintas fases del proceso de arranque del sistema (bootstrap).
themes	Aquí se encuentran todos los temas que vienen con la distribución de Drupal.
modules	Contiene todos los módulos del núcleo que permiten las distintas funcionalidades del CMS, cada uno en su carpeta correspondiente. Nunca se deben modificar directamente los módulos, ni subir módulos adicionales a esta carpeta.



includes	Contiene un conjunto de librerías en forma de archivos PHP con extensión .inc, que incluyen funciones comunes del sistema indispensables para su funcionamiento, como son las conexiones a la base de datos.
DB_GUP	Es la base de datos del GUP de Cuba.
misc	Incluye elementos que tienen que ver con el diseño, y funcionamiento (imágenes, ficheros js).
scripts	Incluye los ficheros de configuración del sistema.
sites	La carpeta sites contiene los extras y modificaciones que se añaden a la distribución original. En esta se encuentran los módulos adicionales creados, añadidos o descargados del repositorio de módulos de Drupal, colocados en sites/all/modules. Además, los temas adicionales instalados o creados, en sites/all/themes.

Para un mejor entendimiento de los módulos utilizados en el desarrollo de la investigación, se ofrece una descripción detallada de los mismos:

- **Button Field:** Agrega un tipo de campo de botón para su uso con cualquier tipo de entidad seleccionable y reglas. Proporciona dos widgets diferentes: Botón HTML y Botón de imagen. Agrega un nuevo evento de clic de botón, una comparación de botón pulsado y una acción de la página de recarga para su uso con reglas. Cuando se hace clic en el botón, se utiliza una llamada AJAX para ejecutar el evento. Utilizando reglas, puede mostrar un mensaje al usuario, modificar algunos datos de la entidad (u otra entidad) y volver a cargar la página actual (o cualquier otra). Este módulo se utiliza para crear botones que permitan la solicitud de matrícula y la inscripción de los eventos.
- **Conditional Fields:** Permite administrar conjuntos de dependencias entre campos. Cuando un campo es "dependiente", sólo estará disponible para editarlo y mostrarse si el estado del campo "dependiente" coincide con la condición correcta. Al editar un nodo (o cualquier otro tipo de entidad que admita campos, como usuarios y categorías). Ejemplo del uso de este módulo se evidencia cuando se selecciona la forma de participación "Ponente" se activa el campo de "Subir ponencia".
- **Date:** Este módulo contiene una fecha y hora flexible de tipo de campo y un API de fecha que otros módulos pueden utilizar. Ejemplo de este en el portal se evidencia en la fecha de inscripción de los eventos y la fecha de matrícula de los cursos.
- **Download File:** Es un módulo para descargar archivos o imágenes. Tiene como características cuatro formateadores, es utilizable en plantillas y vistas y tiene un formato accesible. Se utiliza en el portal para la descarga de la documentación y las herramientas
- **Entity:** Este módulo extiende el API de la entidad de Drupal, con el fin de proporcionar una forma unificada para hacer frente a las entidades y sus propiedades. Además, proporciona un controlador CRUD de entidad, que ayuda a simplificar la creación de nuevos tipos de entidad. En el portal se

evidencia cuando creamos una entidad, el uso de este módulo permite la gestión de la misma.

- Fontawesome: brinda variados íconos para el diseño del portal. Se utiliza para los distintos íconos del portal.
- Fontello: Integra el servicio Fontello a través del módulo Icon API y le permite combinar iconos webfonts. Se utiliza para los distintos íconos del portal.
- Link: Proporciona un campo de contenido personalizado estándar para los enlaces. Con este módulo, los enlaces se pueden agregar fácilmente a cualquier tipo de contenido y perfiles e incluyen validación avanzada y diferentes formas de almacenar enlaces internos o externos y URLs. En el portal se evidencia cuando se crea un campo y este a su vez es re direcciona para otra página.
- Rules: Permite a los administradores definir acciones condicionales ejecutadas en función de eventos ocurridos (conocidos como reglas reactivas). Es un reemplazo con más funciones para el módulo de disparo en núcleo. El uso de este módulo en el portal se refleja cuando el administrador aprueba una solicitud de un usuario, el sistema manda un correo al usuario de forma automática.
- UUID: Este módulo proporciona una API para agregar identificadores universalmente únicos (UUID) a objetos de Drupal, especialmente entidades. Permite obtener los identificadores de los tipos de contenido y así crear dependencias con estos.
- Views: Un módulo contribuido, que permite a los desarrolladores una simple interfaz gráfica para modificar la presentación de contenidos. View permite la selección de campos específicos para mostrar, filtración de diversos atributos de nodo, la elección de opciones de diseño básico, y otras funciones más avanzadas. Muchos desarrolladores de Drupal consideran este módulo uno de los más esenciales para la creación del sitio en general. Se evidencia en cada vista del portal que ha sido publicada.

Los módulos descritos anteriormente son los esenciales para el cumplimiento de las principales funcionalidades del GUP.cu, pero a la vez algunos de ellos necesitan de otros para su correcto funcionamiento, estos módulos adicionales se reflejan en la siguiente tabla:

Tabla 7. dependencia entre los principales módulos utilizados.

Módulos utilizados	Módulos de los que depende
Button Field	Field, Field SQL storage, Rules, Entity tokens, Entity API
Conditional Fields	System, Field, Field SQL storage
Date	Date API
Download File	File, Field, Field SQL storage
Entity	Entity tokens, Rules, Button Field, Rules UI

Fontawesome	Libraries, System
Fontello	Icon API
Rules	Entity tokens, Entity API
UUID	Universally Unique ID, Node, User
Views	Chaos tools

### 3.3. Diseño de la base de datos

El diseño de la base de datos es de gran importancia para el almacenamiento de los datos y para permitir a los usuarios recuperarla y actualizarla en base a sus peticiones posteriormente.

El modelo de datos describe la representación lógica y física de los datos persistentes en el sistema, es frecuentemente necesitado cuando se tiene un modelo de objetos y el mecanismo de almacenamiento se basa en una base de datos relacional. El modelo de datos del sistema tiene baja complejidad, debido a que está desarrollado en el CMS Drupal con un alto número de entidades manejadas por el sistema, por lo que no se contemplarán todas en el modelo.

A continuación, la figura 8 muestra el modelo de datos entidad-relación del sistema para tener una mejor idea de las principales entidades. En el mismo se ve reflejado que en la entidad *node* se guardan los contenidos una vez creados, guardando el tipo de contenido, los comentarios hechos al contenido, el estado del contenido (disponible, cerrado, oculto, entre otros.). la entidad *node\_type* tiene una relación de uno a mucho con *node*, la cual almacena los datos relacionados a cada uno de los tipos de contenidos (descripción, nombre, módulos que usa, si está disponible o no, si el tipo de contenido está bloqueado, entre otros.). También se tiene la tabla *comment*, esta guarda los comentarios hechos en los contenidos por cada usuario, por ello presenta una relación de uno a mucho con la tabla *node* y con la tabla *user*, la tabla *user* guarda todos los datos de los usuarios del sistema y tiene una estrecha relación con la tabla *role* la que almacena el rol que tiene cada usuario, en dependencia del rol que tenga el usuario serán sus permisos (administrador, colaborador, miembro, invitado, entre otros.).

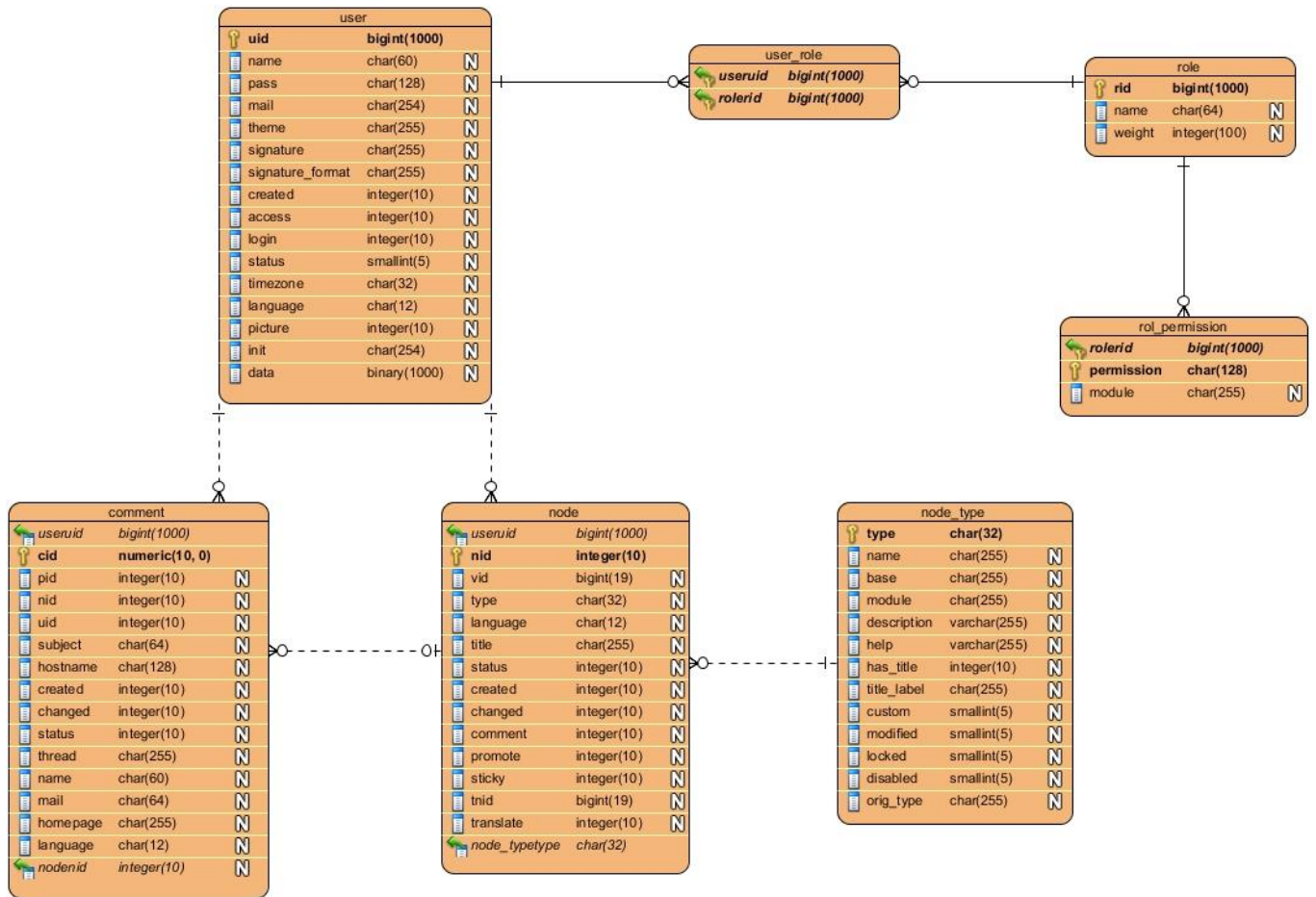


Figura 8. Modelo de Datos Entidad-Relación.

### 3.4. Pantalla principal del sistema

La pantalla principal del portal para la gestión del Grupo de Usuarios PostgreSQL de Cuba es el resultado visual donde se define cada una de las páginas del sistema, ubicando en ellas todos los elementos que las componen. En esta se tiene un menú principal donde se muestra las distintas páginas, cuenta también con un *slide* donde se muestran las 3 noticias más relevantes del GUP.cu, se tiene un panel donde se muestra un enlace hacia los cursos, eventos y la comunidad, esta página muestra también las últimas noticias publicadas.



Figura 9. Pantalla principal del GUP de Cuba.

### 3.5. Pruebas realizadas al sistema

Las pruebas de software comprenden una fase muy importante del proceso de desarrollo. Tienen como objetivo determinar y asegurar la calidad, fiabilidad y robustez de un sistema, dentro del contexto o escenario donde está previsto que sea utilizado. Además, permiten encontrar el mayor número de errores con la menor cantidad de tiempo y esfuerzo posible. Por otra parte, muestran si las funcionalidades del software operan según las especificaciones y requisitos definidos (Mateo, y otros, 2009).

Debido a que las pruebas son aplicadas para diferentes tipos de objetivos, en diferentes escenarios o niveles de trabajos, existen varios niveles de prueba (desarrollador, independiente, unidad, integración, sistema y aceptación). En la estrategia de prueba a seguir se realizarán pruebas a nivel de sistema y de aceptación.

Las pruebas de sistema son aquellas que se hacen cuando el software está funcionando como un todo. Están dirigidas a verificar el programa final después que todos los componentes de software y hardware han sido integrados. A este nivel se realizarán pruebas de funcionalidad, rendimiento y fiabilidad.

Las pruebas de aceptación son las que se realizan al final antes del despliegue del sistema. Su objetivo es verificar que el software está listo y que puede ser usado por usuarios finales para ejecutar aquellas

funciones y tareas para las cuales el software fue construido. Son generalmente desarrollada y ejecutada por el cliente o un especialista de la aplicación (García, 2013). A este nivel se realizarán pruebas de función. Los tipos de pruebas seleccionados para validar el correcto funcionamiento del portal a nivel de sistema son los siguientes:

- Pruebas funcionales: están enfocadas a realizar pruebas de función y seguridad. Estas pruebas son muy útiles ya que permiten comprobar los requisitos definidos contra las funcionalidades presentes en la aplicación, asegurar que el sistema es accedido por los actores definidos según niveles de acceso.
- Pruebas de rendimiento: dentro de las pruebas de rendimiento se realizarán pruebas de cargas para validar y valorar la aceptabilidad de los límites operacionales de un sistema bajo carga de trabajo variable, mientras el sistema bajo prueba permanece constante. La variación en carga es simular la carga de trabajo promedio y con los picos que ocurre dentro de tolerancias operacionales normales.
- Pruebas de fiabilidad: dentro de estas pruebas se encuentra las de estrés, la cual está enfocada a evaluar cómo el sistema responde bajo condiciones anormales, (extrema sobrecarga, insuficiente memoria, servicios y hardware no disponible).

### **3.5.1. Métodos de prueba**

Existen fundamentalmente dos métodos de prueba, el de caja blanca y el de caja negra.

Las pruebas de caja blanca requieren del conocimiento de la estructura interna del programa y son derivadas a partir de las acciones internas de diseño o del código. Mediante los métodos de prueba de la caja blanca, se puede obtener casos de prueba que garanticen que:

- Se ejerciten por lo menos una vez todos los caminos independientes para cada módulo.
- Se ejerciten todas las decisiones lógicas en sus vertientes verdaderas y falsas.
- Ejecuten todos los bucles en sus límites y con sus límites operacionales.
- Se ejerciten las estructuras internas de datos para asegurar su validez.

Se denomina caja negra a aquel elemento que es estudiado desde el punto de vista de las entradas que recibe y las salidas o respuestas que produce. Las pruebas de caja negra se llevan a cabo sobre la interfaz del software, para obviar el comportamiento interno y la estructura del programa (García, 2013). Las pruebas de caja negra no son una alternativa a las técnicas de pruebas de caja blanca, más bien se trata de un enfoque complementario que intenta descubrir diferentes tipos de errores (Pressman, 2010).

Los casos de prueba de la caja negra pretenden demostrar que:

- Las funciones del software son operativas.

- La entrada se acepta de forma correcta.
- Se produce una salida correcta.
- La integridad de la información externa se mantiene.

Dentro del método de caja negra se aplicará la técnica Partición de Equivalencia, que divide el campo de entrada en clases de datos que tienden a ejecutar determinadas funciones del software (Pressman, 2010).

### **3.5.2. Diseño de los casos de prueba**

Un caso de prueba es un conjunto de entradas de pruebas, condiciones de ejecución y resultados esperados desarrollados para cumplir un objetivo en particular o una función esperada, la entidad más simple que siempre es ejecutada como una unidad, desde el comienzo hasta el final. Los casos de pruebas deben verificar si el producto satisface los requerimientos del usuario, tal y como se describe en las especificaciones de los requisitos y deben verificar si el producto se comporta como desea, tal y como se describe en las especificaciones funcionales del diseño (Sommerville, 2005).

Las siguientes tablas muestran los diseños de caso de prueba para el CU Gestionar Evento.

Tabla 8. Sección Insertar Evento del CU Gestionar Eventos.

Escenario	Descripción	Nombre	Descripción	Fecha	Lugar	Respuesta del sistema	Flujo central
EC 1.1: Insertar nuevo evento.	Permite insertar un nuevo evento.	V	V	V	V	Se inserta el evento y se muestra un mensaje "Se ha insertado un evento" y se actualiza el listado de eventos.	1-Agregar Contenido. 2- Seleccionar tipo de contenido "Eventos". 3-Se introducen los datos. 4-Se presiona el botón "Guardar".
		Pgday 2016	Evento importante para la comunidad	16/02/2016 - 20/02/2016	UCI		
EC 1.2: Cancelar opción insertar nuevo evento.	Permite cancelar la opción insertar un nuevo evento.	V	V	V	V	Se cancela la acción insertar nuevo evento.	1-Agregar Contenido. 2- Seleccionar tipo de contenido "Eventos". 3-Se introducen los datos. 4-Se presiona el botón "Cerrar".
		Pgday 2016	Evento importante para la comunidad	16/02/2016 - 20/02/2016	UCI		
EC 1.3: Insertar un evento con un campo obligatorio vacío	Verifica si se introdujo todos los valores de los campos obligatorios.	-	V	V	V	No se crea el evento y se muestra un mensaje "El campo Nombre de Evento es obligatorio", resaltando el campo vacío.	1-Agregar Contenido. 2- Seleccionar tipo de contenido "Eventos". 3-Se introducen los datos. 4-Se presiona el botón "Guardar".
		-	Evento importante para la comunidad	16/02/2016 - 20/02/2016	UCI		
EC 1.4: Introducir un rango de fecha no válido.	Verifica si se introdujo un rango de fecha válido.	V	V	V	V	No se crea el evento y se muestra un mensaje "El rango de fecha no es válido", resaltando el campo fecha.	1-Agregar Contenido. 2- Seleccionar tipo de contenido "Eventos". 3-Se introducen los datos. 4-Se presiona el botón "Guardar".
		Pgday 2016	Evento importante para la comunidad	25/02/2016 - 20/02/2016	UCI		



Tabla 9. Sección Modificar Evento del CU Gestionar Eventos.

Escenario	Descripción	Nombre	Descripción	Fecha	Lugar	Respuesta del sistema	Flujo central
EC 2.1: Modificar evento.	Permite modificar un evento.	V	V	V	V	Se modifica el evento seleccionado y se muestra un mensaje "El evento ha sido actualizado" y se actualiza el evento.	1-Buscar contenido. 2- Seleccionar tipo de contenido a modificar. 3- Se modifican los datos. 4- Se presiona el botón "Guardar".
		PgAdmin 2016	Evento más importante para la comunidad	16/02/2016 - 20/02/2016	UCI		
EC 2.2: Cancelar opción modificar evento.	Permite cancelar la opción modificar evento.	V	V	V	V	Se cancela la acción modificar evento.	1-Buscar contenido. 2- Seleccionar tipo de contenido a modificar. 3- Se modifican los datos. 4- Se presiona el botón "Cerrar".
		Pgday 2016	Evento importante para la comunidad	16/02/2016 - 20/02/2016	UCI		
EC 2.3: Modificar un evento con un campo obligatorio vacío	Verifica si se introdujo todos los datos de los campos obligatorios.	-	V	V	V	No se modifica el evento y se muestra un mensaje "El campo Nombre de Evento es obligatorio", resaltando el campo vacío.	1-Buscar contenido. 2- Seleccionar tipo de contenido a modificar. 3- Se modifican los datos. 4- Se presiona el botón "Cerrar".
		-	Evento importante para la comunidad	16/02/2016 - 20/02/2016	UCI		
EC 2.4: Introducir un rango de fecha no válido.	Verifica si se introdujo un rango de fecha válido.	V	V	V	V	No se crea el evento y se muestra un mensaje "El rango de fecha no es válido", resaltando el campo fecha.	1-Agregar Contenido. 2- Seleccionar tipo de contenido "Eventos". 3-Se introducen los datos. 4-Se presiona el botón "Guardar".
		Pgday 2016	Evento importante para la comunidad	25/02/2016 - 20/02/2016	UCI		

Tabla 10. Sección Eliminar Evento del CU Gestionar Eventos.

Escenario	Descripción	Nombre	Descripción	Fecha	Lugar	Respuesta del sistema	Flujo central
EC 3.1: Eliminar evento.	Permite eliminar un evento.	N/A	N/A	N/A	N/A	Se elimina el evento seleccionado y se muestra un mensaje "¿Está seguro de que quiere eliminar "Evento"?" y si se confirma se elimina el evento.	1-Buscar contenido. 2- Seleccionar tipo de contenido a eliminar. 3-Se presiona el botón "eliminar". 4-Se confirma presionando el botón "Eliminar".
		N/A	N/A	N/A	N/A		
EC 2.2: Cancelar opción eliminar evento.	Permite cancelar la opción eliminar evento.	N/A	N/A	N/A	N/A	Se cancela la acción eliminar evento.	1-Buscar contenido. 2- Seleccionar tipo de contenido a eliminar. 3-Se presiona el botón "eliminar". 4-Se cancela presionando el botón "Cancelar".
		N/A	N/A	N/A	N/A		

Tabla 11. Sección Buscar Evento del CU Gestionar Eventos.

Escenario	Descripción	Nombre	Descripción	Fecha	Lugar	Respuesta del sistema	Flujo central
EC 4.1: Buscar evento.	Permite buscar un evento.	N/A	N/A	N/A	N/A	Se busca un evento.	1-Buscar contenido. 2-Se muestran todos los contenidos.
		N/A	N/A	N/A	N/A		

Tabla 12. Diseño de caso de prueba para pruebas de Rendimiento.

ID del Escenario	Escenarios de la sección	Carga de trabajo	Descripción	Resultado esperado	Resultado
EC 1: Acceder a la interfaz principal del GUP.cu.	EC 1.1: Acceder a la interfaz principal del GUP.cu (localhost/tesis )	50	Esta interfaz es el inicio, por tanto, es la primera por la que accederán todos los usuarios del sistema. Es el puente hacia el resto de las páginas que componen el sitio.	La aplicación tiene que responder en un tiempo menor que 5 seg.	3,5 seg
EC 2: Eventos	EC 2.1: Mostrar Eventos. Para ello se da clic en la pestaña Eventos.	100	Esta interfaz muestra toda la información relacionada con los "Eventos".	La aplicación tiene que responder en un tiempo menor que 5 seg.	4,0 seg
EC 3: Cursos	EC 3.1: Mostrar Cursos Para ello se da clic en la pestaña Cursos.	150	Esta interfaz muestra toda la información relacionada con los "Cursos".	La aplicación tiene que responder en un tiempo menor que 5 seg.	4,2 seg

### 3.5.3. Resultados de las pruebas

La aplicación de la estrategia de prueba definida arrojó los siguientes resultados.

#### Resultado de las pruebas funcionales

Las pruebas funcionales aplicadas a la propuesta de solución son basadas en el diseño de casos de pruebas, donde se recogen los escenarios correspondientes a cada requisito funcional especificado. Para la realización de las pruebas se ejecuta cada caso de prueba utilizando datos válidos e inválidos y se eligen correctamente los valores de entrada abarcando el mayor número posible de combinaciones, sin que sea muy elevado el número de casos de pruebas.

En la primera iteración se detectaron 14 no conformidades, de ellas 6 no conformidades son de error de idioma, 5 no conformidades en los RF22 y RF23 y 3 no conformidades de errores funcionales en los RF25 y RF26. En la segunda iteración se encontró un total de 4 no conformidades en los RF32 y RF33. Todas las no conformidades encontradas fueron solucionadas por el equipo de desarrollo como se muestra en la siguiente tabla.

Tabla 13. Resultado de las iteraciones de las pruebas funcionales.

Iteración	No Conformidades	Requisitos Funcionales	Solución de las no conformidades
1	14	RF22, RF23, RF25 y RF26	14
2	4	RF32 y RF33	4
3	0	-	0

#### Resultado de las pruebas de carga y estrés

Se ejecutaron una serie de funcionalidades que brinda el sistema con el objetivo de evaluar el tiempo de respuesta del proceso de gestionar evento. La ejecución de las pruebas se realizó sobre el entorno de trabajo que presenta como características: Microprocesador Intel Core i5 a 2,20 Ghz con 8 GB de RAM. Se utilizó la herramienta de software libre, JMeter, que ofrece la posibilidad de realizar pruebas de carga sobre diferentes aspectos de una aplicación web, inicialmente diseñada para pruebas de estrés en las mismas. Utilizar JMeter supone una mayor efectividad en el proceso y en la fiabilidad de los resultados. Dispone de varios componentes que facilitan la elaboración de los escenarios de prueba con la ventaja de simular para cada uno de esos escenarios miles de usuarios. De esta manera se verifica el rendimiento del sistema mediante las pruebas de carga y estrés. Las pruebas arrojaron como resultados que para la cantidad promedio esperada de usuarios conectados concurrentemente (entre 50 y 100) se obtuvieron tiempos de

respuesta entre 3 y 4 segundos, valorados como aceptables, mientras que más de 100 peticiones concurrentes, la aplicación mantuvo su correcto funcionamiento.

Tabla 14. Resultados de las pruebas de rendimiento.

<b>Usuarios</b>	<b>Error %</b>	<b>Rendimiento (tiempo de respuesta)</b>
<b>50</b>	0,00	3,5 /seg
<b>100</b>	0,00	4,0 /seg
<b>150</b>	0,00	4,2 /seg

## **Conclusiones del capítulo**

En el capítulo se realizó el diagrama de despliegue, formado por un cliente, un servidor web y uno de bases de datos; se realizó la vista de implementación, formado por 12 componentes y; se definió la estrategia para la verificación de la solución, realizándose pruebas de caja negra, para la que fueron definidos los diseños de casos de prueba, una vez aplicados, permitieron la detección de 14 no conformidades, resueltas en 3 iteraciones, se realizaron pruebas de rendimiento, dentro de ellas las de carga y estrés, que con el uso de la herramienta Jmeter 2.3.1 permitió medir el tiempo de respuesta de varias solicitudes hechas por 100 usuarios conectados concurrentemente y el portal mantuvo un funcionamiento correcto.

## CONCLUSIONES GENERALES

---

Con la realización del trabajo de diploma se logró cumplir con los objetivos planteados, desarrollándose una solución que ofrece a los miembros del GUP.cu la posibilidad de gestionar sus miembros, eventos y acciones de capacitación mediante un portal web para la comunidad de PostgreSQL de Cuba. Por lo que se concluye que:

- El estudio de los fundamentos teóricos de la investigación evidenció la necesidad de implementar un portal para la gestión de los procesos llevados a cabo por el GUP.cu así como la selección de la metodología y herramientas que cuentan con las características necesarias para desarrollar la solución propuesta.
- El análisis y diseño del portal para la gestión de los procesos llevados a cabo por el GUP.cu permitió un mejor entendimiento del proceso de negocio permitiendo la implementación de una solución que responde a las necesidades del cliente.
- La implementación de la solución permitió obtener como resultado un portal para la gestión de los procesos llevados a cabo por el Grupo de Usuarios PostgreSQL de Cuba, en el que se gestionan, además de noticias y documentación, los miembros del GUP.cu, así como actividades de capacitación y eventos.
- La validación de la solución mediante la aplicación de los casos de pruebas, las pruebas de aceptación y rendimiento permitió obtener un portal web funcional que cumple con los requisitos especificados por el cliente.

## **RECOMENDACIONES**

---

Independientemente de que se hayan alcanzado los objetivos trazados al inicio de la investigación se recomienda:

- Agregar funcionalidades para la revisión y seguimiento de trabajos para eventos.
- Agregar funcionalidades para la notificación de notas en las actividades de capacitación.

## BIBLIOGRAFÍA

---

- ALEGSA. 2016.** ALEGSA.com.ar. *ALEGSA.com.ar*. [En línea] 2016. [Citado el: 16 de marzo de 2017.] <http://www.alegsa.com.ar/Dic/servidor.php>.
- Alvarez de Zayas, Carlos. 1995.** *METODOLOGIA DE LA INVESTIGACION CIENTIFICA*. Santiago de Cuba : s.n., 1995.
- Alvarez, Sara. 2007.** desarrolloweb. *desarrolloweb*. [En línea] 31 de julio de 2007. <http://www.desarrolloweb.com/articulos/sistemas-gestores-bases-datos.html>.
- Apache. 2016.** The Apache Software Foundation. *The Apache Software Foundation*. [En línea] 2016. [Citado el: 16 de marzo de 2017.] <https://www.apache.org/>.
- ArPUG. 2008.** Grupo de Usuarios PostgreSQL de Argentina. *Grupo de Usuarios PostgreSQL de Argentina*. [En línea] 22 de noviembre de 2008. [Citado el: 11 de diciembre de 2016.] <http://www.postgresql.org.ar>.
- Avison, D. y Fitzgerald, G. 1995.** *Information Systems Development: Methodologies, Techniques and Tools*. s.l. : McGraw-Hill, 1995.
- Ayen, J. 2012.** *Drupal 7 Views Cookbook*. Birmingham : Packt Publishing Ltd., 2012. ISBN 978-1-84951-434-7.
- Barrientos, Pablo A. 2014.** SEDICI Repositorio Institucional de la UNLP. *SEDICI Repositorio Institucional de la UNLP*. [En línea] 2014. [Citado el: 11 de mayo de 2017.] <http://sedici.unlp.edu.ar/handle/10915/34969>.
- Cabrera, Lianet y Pompa, Enrique R. 2012.** *Extensión de Visual Paradigm for UML para el desarrollo dirigido por modelos de aplicaciones de gestión de información*. 2012. pág. 6.
- Carrillo, Isaías, Pérez, Rodrigo y Rodríguez, Aureliano. 2008.** tecnicaytecnologiasc. *tecnicaytecnologiasc*. [En línea] 15 de octubre de 2008. [Citado el: 13 de enero de 2017.] <https://tecnicaytecnologiasc.wikispaces.com/file/view/Metodologias+de+desarrollo.pdf>.
- Carvajal, Jose Carlos. 2008.** *Metodologías Ágiles: Herramientas y modelo de desarrollo para aplicaciones Java EE como metodología empresarial*. Barcelona : s.n., 2008.
- Comunidad de PostgreSQL Brasileña. 2015.** Comunidad de PostgreSQL Brasileña. *Comunidad de PostgreSQL Brasileña*. [En línea] 2015. [Citado el: 15 de marzo de 2017.] <http://postgresql.org.br>.
- Concepto. 2015.** concepto.de. *concepto.de*. [En línea] 2015. [Citado el: 15 de marzo de 2017.] <http://concepto.de/comunidad/>.
- Córcoles, César y Casado, Carlos. 2008.** *Crear y mantener una página web*. 2008.



**Cuerda, Xavier. 2004.** *Introducción a los Sistemas de Gestión de Contenidos (CMS) de código abierto.* 2004.

**Date, C. J. 1999.** *Introducción a los sistemas de bases de datos.* California : Editorial Félix Varela, 1999.

**definicion. 2017.** definicion.de. *definicion.de.* [En línea] 2017. [Citado el: 15 de marzo de 2017.] <http://definicion.de/html/>.

—. **2016.** definicion.org. *definicion.org.* [En línea] 10 de noviembre de 2016. [Citado el: 16 de marzo de 2017.] <http://www.definicion.org/lenguaje-de-programacion>.

**desarrolloweb. 2012.** desarrolloweb.com. *desarrolloweb.com.* [En línea] 2012. [Citado el: 16 de marzo de 2017.] <https://desarrolloweb.com/javascript/>.

**Domínguez, Yanetsys y González, Pedro. 2006.** Scielo. *Scielo.* [En línea] julio-agosto de 2006. [Citado el: 14 de enero de 2017.] [http://scielo.sld.cu/scielo.php?script=sci\\_arttext&pid=S1024-94352006000400011](http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1024-94352006000400011).

**Drupal. 2017.** Drupal. *Drupal.* [En línea] 2017. [Citado el: 15 de marzo de 2017.] <https://www.drupal.org>.

—. **2016.** Drupal. *Drupal.* [En línea] 23 de diciembre de 2016. [Citado el: 30 de marzo de 2017.] <https://www.drupal.org/node/547518>.

**Drupal Hispano. 2005.** Drupal Hispano. Comunidad de usuarios de Drupal. *Drupal Hispano. Comunidad de usuarios de Drupal.* [En línea] 11 de abril de 2005. [Citado el: 15 de marzo de 2017.] <http://drupal.org.es/>.

**Drupal. 2017.** Tutorial Drupal. *Tutorial Drupal.* [En línea] 2017. [Citado el: 30 de marzo de 2017.] <http://www.cursosdrupal.com/content/arquitectura>.

**Epíscopo, Daniel A. 2008.** Seguridad en ambientes laborales. *Seguridad en ambientes laborales.* [En línea] 8 de octubre de 2008. [Citado el: 13 de diciembre de 2016.] <https://seguridadyambiente.wordpress.com/2008/10/11/ventajas-de-trabajar-bajo-un-sistema-de-gestion/>.

**Fables, Ailyn. 2011.** RCCI. *RCCI.* [En línea] 2011. [Citado el: 24 de abril de 2017.] [http://rcci.uci.cu/index.php?journal=rcci&page=article&op=view&path\[\]=84&path\[\]=72](http://rcci.uci.cu/index.php?journal=rcci&page=article&op=view&path[]=84&path[]=72).

**Falgueras, Benet. 2002.** *Ingeniería del software.* s.l. : UOC, 2002. 8484297934.

**García, Marisol. 2013.** *Ingeniería de software 2.* [Presentación electrónica] 2013.

**Gil, Francisco. 2012.** *Experto en Drupal 7. Curso de creación y gestión de portales web con Drupal 7.* s.l. : Forcontu SL, 2012. 8493941018, 9788493941017.

**González, José E. 2008.** docIRS. *docIRS.* [En línea] enero de 2008. [Citado el: 4 de abril de 2017.] <http://www.docirs.com/uml.htm>.

**Ibiblio. 2017.** Modelado de Sistemas com UML. *Modelado de Sistemas com UML*. [En línea] 2017. [Citado el: 30 de marzo de 2017.] <http://www.ibiblio.org/pub/linux/docs/LuCaS/Tutoriales/doc-modelado-sistemas-UML/multiple-html/x219.html>.

**James, Dylan . 2012.** *Drupal 7 Cookbook. Over 70 recipes that will advance your Drupal skills from novice to pro*. Birmingham : Packt Publishing Ltd., 2012. ISBN 978-1-84951-796-6.

**JLAYA. 2016.** jlaya.com. *jlaya.com*. [En línea] 29 de julio de 2016. [Citado el: 23 de octubre de 2016.] <http://www.jlaya.com/servidores-web-mas-utilizados/>.

**Joomla. 2017.** Joomla. *Joomla*. [En línea] 2017. [Citado el: 15 de marzo de 2017.] <https://www.joomla.org>.

**JS. 2016.** JavaScript.com. *JavaScript.com*. [En línea] 2016. [Citado el: 16 de marzo de 2017.] <https://www.javascript.com/>.

**Lamarca, María J. 2013.** HTML. *HTML*. [En línea] 8 de diciembre de 2013. [Citado el: 15 de marzo de 2017.] <http://www.hipertexto.info/documentos/html.htm>.

**Larman, Craig. 2004.** *UML y Patrones*. La Habana : Felix Varela, 2004.

**Madel, Kurt . 2012.** *Drupal 7 Development by Example. Beginner's Guide*. Birmingham : Packt Publishing Ltd, 2012. ISBN 978-1-84951-680-8.

**Martínez, Aymeé y Elías, Martín. 2012.** *Análisis, diseño e implementación de funcionalidades para el Portal web Preparación Para la Defensa de la Facultad 4*. La Habana : s.n., 2012.

**Martinez, Rafael. 2010.** postgresql-es. *postgresql-es*. [En línea] 2 de octubre de 2010. [Citado el: 16 de marzo de 2017.] [www.postgresql.org/es/sobre\\_postgresql](http://www.postgresql.org/es/sobre_postgresql).

**Mateo, Pedro, Martínez, Gregorio y Sevilla, Diego. 2009.** redalyc.org. *redalyc.org*. [En línea] 2009. [Citado el: 24 de abril de 2017.] <http://www.redalyc.org/articulo.oa?id=92217159003>. E-ISSN: 1885-4486.

**Mendivelso, Luis F. 2013.** Instartius Corporation. *Instartius Corporation*. [En línea] 2013. [Citado el: 13 de marzo de 2017.] [http://instartius.com/site/?page\\_id=211](http://instartius.com/site/?page_id=211).

**Mendoza, Alberto. 2009.** *Definición de la arquitectura de software del Grupo de Desarrollo para la Gestión de Equipos Médicos*. La Habana : s.n., 2009.

**Mercer, David. 2010.** *Drupal 7. Create and operate any type of website quickly*. s.l. : Packt Publishing Ltd. 32 Lincoln Road, 2010. ISBN 978-1-849512-86-2.

**Mesa, Yunior y Vázquez, Yudisney. 2012.** *Espacio de comunicación e intercambio para la Comunidad Técnica*. 2012. pág. 3.

**Microsoft. 2016.** Microsoft. *Microsoft*. [En línea] 10 de noviembre de 2016. [Citado el: 17 de marzo de 2017.] [https://msdn.microsoft.com/es-es/library/hh831725\(v=ws.11\).aspx](https://msdn.microsoft.com/es-es/library/hh831725(v=ws.11).aspx).

**MisRespuestas. 2016.** MisRespuestas.com. *MisRespuestas.com*. [En línea] 7 de Noviembre de 2016. [Citado el: 17 de marzo de 2017.] [www.misrespuestas.com/que-es-un-servidor-web.html](http://www.misrespuestas.com/que-es-un-servidor-web.html).

**NetBeans. 2012.** Bienvenido a NetBeans. *Bienvenido a NetBeans*. [En línea] diciembre de 2012. [Citado el: 12 de enero de 2017.] [http://netbeans.org/index\\_es.html](http://netbeans.org/index_es.html).

**Nevado, Ma Victoria. 2010.** *Introducción a las bases de datos relacionales*. Madrid : Vision Libros, 2010.

**Nipas, Gustavo. 2014.** SlideShare. *SlideShare*. [En línea] 27 de octubre de 2014. [Citado el: 14 de marzo de 2017.] <http://es.slideshare.net/nipas/10-sgbd>.

**openSUSE. 2012.** openSUSE.org. *openSUSE.org*. [En línea] 2012. [Citado el: 23 de marzo de 2017.] <https://es.opensuse.org/Apache>.

**Pavón, Juan. 2009.** Portal FDI. Universidad Complutense de Madrid. *Portal FDI. Universidad Complutense de Madrid*. [En línea] 2009. [Citado el: 21 de marzo de 2017.] <https://www.fdi.ucm.es/profesor/jpavon/poo/2.14.MVC.pdf>.

**Peacock, Michael . 2011.** *Drupal 7 Social Networking*. Birmingham : Packt Publishing Ltd., 2011. ISBN 978-1-849516-00-6.

**Pérez, Adrián. 2012.** *Guía básica de Drupal 7*. La Habana : s.n., 2012.

**php. 2001.** php. *php*. [En línea] 2001. [Citado el: 12 de diciembre de 2016.] <http://php.net/manual/es/intro-what-is.php>.

**Ponce, J., y otros. 2010.** REICIS. Revista Española de Innovación, Calidad e Ingeniería del Software. *REICIS. Revista Española de Innovación, Calidad e Ingeniería del Software*. [En línea] junio de 2010. [Citado el: 11 de mayo de 2017.] <http://www.redalyc.org/articulo.oa?id=92218768004>. E-ISSN:1885-4486.

**Pons, Olga, y otros. 2009.** *Introducción a las Bases de Datos: el modelo relacional*. Madrid : Thomson Editors Spain, 2009.

**PostgreSQL. 2017.** The PostgreSQL Global Development Group. *The PostgreSQL Global Development Group*. [En línea] 2017. [Citado el: 21 de marzo de 2017.] <https://www.postgresql.org/files/documentation/pdf/9.6/postgresql-9.6-A4.pdf>.

**PostgreSQL.it. 2009.** PostgreSQL.it. *PostgreSQL.it*. [En línea] 1 de Diciembre de 2009. <http://www.psql.it>.

**PostgreSQL-es. 2013.** PostgreSQL-es. Portal en español sobre PostgreSQL. *PostgreSQL-es. Portal en español sobre PostgreSQL*. [En línea] 2013. [Citado el: 15 de marzo de 2017.] <http://www.postgresql.org/es/>.

**Pressman, Roger S. 2010.** *Ingeniería del Software. Un enfoque práctico.* México D. F. : McGRAW-HILL INTERAMERICANA EDITORES, 2010. ISBN: 978-607-15-0314-5.

**Programacion. 2016.** Programacion.net. *Programacion.net.* [En línea] 14 de diciembre de 2016. [http://programacion.net/articulo/por\\_que\\_elegir\\_php\\_143](http://programacion.net/articulo/por_que_elegir_php_143).

**Programando la web. 2016.** Programando la web. *Programando la web.* [En línea] 2016. <http://www.programandolaweb.com/posts/php/6/Ventajas-y-desventajas-en-el-uso-de-php.html>.

**Reyna, Rafael. 2012.** scribd. *scribd.* [En línea] 2012. [Citado el: 3 de febrero de 2017.] <https://es.scribd.com/document/3062020/Capitulo-I-HERRAMIENTAS-CASE>.

**Robert, J. y Pakrul, Stephanie. 2010.** *Foundation Drupal 7.* 2010. ISBN-13 (pbk): 978-1-4302-2808-0.

**Romero, Hermenegildo. 2012.** SlideShare. *SlideShare.* [En línea] 7 de febrero de 2012. [Citado el: 15 de mayo de 2017.] [https://es.slideshare.net/MeneRomero/metodologias-de-desarrollo?qid=b446e404-99e5-490c-9d39-1256960937e0&v=&b=&from\\_search=1](https://es.slideshare.net/MeneRomero/metodologias-de-desarrollo?qid=b446e404-99e5-490c-9d39-1256960937e0&v=&b=&from_search=1).

**Rosanigo, Zulema. 2000.** Postgrado. Facultad de Informática. *Postgrado. Facultad de Informática.* [En línea] febrero de 2000. [Citado el: 30 de marzo de 2017.] [http://postgrado.info.unlp.edu.ar/Carreras/Magisters/Ingenieria\\_de\\_Software/Tesis/Rosanigo\\_Zulema.pdf](http://postgrado.info.unlp.edu.ar/Carreras/Magisters/Ingenieria_de_Software/Tesis/Rosanigo_Zulema.pdf).

**Rumbaugh, James, Jacobson, Ivar y Booch, Grady. 1999.** *El lenguaje unificado de modelado. Manual de referencia.* Madrid : Addison Wesley, 1999. ISBN: 84-7829-037-0.

**Sánchez, Tamara. 2015.** *Metodología de desarrollo para la Actividad productiva de la UCI.* La Habana : s.n., 2015.

**Selva, I. 2014.** SCRIBD. *SCRIBD.* [En línea] 2014. [Citado el: 15 de mayo de 2017.] <https://es.scribd.com/document/294216411/aup-herramientas>.

**Sierra, Manuel. 2016.** aprenderaprogramar.com. *aprenderaprogramar.com.* [En línea] 2016. [Citado el: 12 de febrero de 2017.] [http://aprenderaprogramar.com/index.php?option=com\\_content&view=article&id=542:que-es-un-servidor-y-cuales-son-los-principales-tipos-de-servidores-proxydns-webftppop3-y-smtp-dhcp&catid=57:herramientas-informaticas&Itemid=179](http://aprenderaprogramar.com/index.php?option=com_content&view=article&id=542:que-es-un-servidor-y-cuales-son-los-principales-tipos-de-servidores-proxydns-webftppop3-y-smtp-dhcp&catid=57:herramientas-informaticas&Itemid=179).

**SINCAL. 2013.** Servicios y Sistemas Integrales de Capacitación y Calidad. *Servicios y Sistemas Integrales de Capacitación y Calidad.* [En línea] 6 de Diciembre de 2013. <https://sincalmx.wordpress.com/2013/12/06/que-es-y-para-que-sirve-un-sistema-de-gestion/>.

**Sommerville, Ian. 2005.** *Ingeniería del Software*. Madrid : Pearson Addison Wesley, 2005. ISBN: 84-7829-074-5.

**Stovall, Tom. 2012.** *Drupal 7 Mobile Web Development. Beginner's Guide*. Birmingham : Packt Publishing Ltd., 2012. ISBN 978-1-84951-562-7.

**taringa. 2009.** Taringa. *Taringa*. [En línea] 13 de julio de 2009. <http://www.taringa.net/posts/info/2688390/Apache-vs-IIS-cual-elegir.html>.

**THINK&SELL. 2016.** THINK&SELL. *THINK&SELL*. [En línea] 2016. [Citado el: 2 de enero de 2017.] <http://thinkandsell.com/servicios/consultoria/software-y-sistemas/sistemas-de-gestion-normalizados/>.

**Tomlinson, Todd y VanDyk, John K. 2010.** *Pro Drupal 7 Development. Third Edition*. New York : s.n., 2010. ISBN-13 (pbk): 978-1-4302-2838-7.

**Turner, Víctor. 2013.** Significados. *Significados*. [En línea] 2013. [Citado el: 12 de febrero de 2017.] <https://www.significados.com/comunidad/>.

**Tuthanh, Le. 2012.** *Master Drupal in 7 hours*. 2012.

**Vázquez, Yudisney. 2013.** Repositorio Institucional de la Universidad de las Ciencias Informáticas. *Repositorio Institucional de la Universidad de las Ciencias Informáticas*. [En línea] 29 de Octubre de 2013. [Citado el: 12 de octubre de 2017.] <https://repositorio.uci.cu/jspui/handle/ident/7996>.

**Vázquez, Yudisney, Castillo, Gilberto y Mier, Lisleydi. 2011.** *Propuesta de un plan de capacitación para la preparación y futura certificación en PostgreSQL*. 2011. págs. 2-3.

**W3C. 2017.** Cascading Style Sheets. *Cascading Style Sheets*. [En línea] 2017. [Citado el: 15 de marzo de 2017.] <https://www.w3.org/Style/CSS>.

**Welling, Luke y Thomson, Laura. 2005.** *Desarrollo Web con PHP y MySQL*. Madrid : AnayaMultimedia, 2005.

**WM. 2009.** elwebmaster. *elwebmaster*. [En línea] 27 de julio de 2009. [Citado el: 13 de marzo de 2017.] <http://www.elwebmaster.com/articulos/los-10-cms-mas-recomendados>.

**wordpress. 2016.** wordpress.org. *wordpress.org*. [En línea] 25 de octubre de 2016. [Citado el: 12 de enero de 2017.] <https://es.wordpress.org/>.



### ACTA DE ACEPTACIÓN

Por medio de la presente se hace constar que el Portal para el Grupo de Usuarios PostgreSQL de Cuba fue entregado con los requerimientos pactados desarrollados, haciéndose entrega formal al Grupo de:

- El Portal.
- Un respaldo de la base de datos.

Y para que conste su aceptación, se firma la presente a los 14 días del mes de Junio de 2017.

Gilberto Castillo Martínez  
Contacto de Prensa del Grupo de Usuarios PostgreSQL de Cuba

Figura 10. Acta de aceptación del GUP.cu.